Lecture 4

Image Segmentation

Dr. Amira Salah Ashour

Image analysis and segmentation

- Detection of Discontinuity
  - Point, line, edge and combined detection..
- Edge linking and boundary detection
  - Local processing, hough transform, graph-theoretic technique..
- Thresholding
  - Global thresholding, Optimal thresholding, threshold selection..
- Region oriented segmentation
  - Region growing, Region splitting and merging..

Introduction

- Image analysis:-

  Techniques for extracting information from an image.

- Segmentation is the first step for image analysis.
- Segmentation is used to subdivide an image into its constituent parts or objects.
- Segmentation:
  - Split or separate an image into regions
  - To facilitate recognition, understanding, and region of interests (ROI) processing
- This step determines the eventual success or failure of image analysis.
- Generally, the segmentation is carried out only up to the objects of interest are isolated. e.g. face detection.
- The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze.

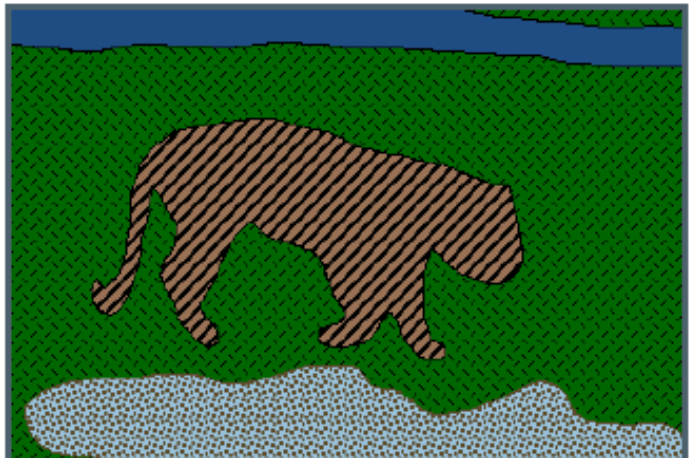
Segmentation

- Image segmentation is the process of partitioning a digital image into multiple parts

- It divide the image into: meaningful and/or perceptually uniform regions

- Segmentation is typically used to locate objects and boundaries of physical entities in the scene

- The segmentation process utilizes available image information (graylevel, colour, texture, pixel position, …).

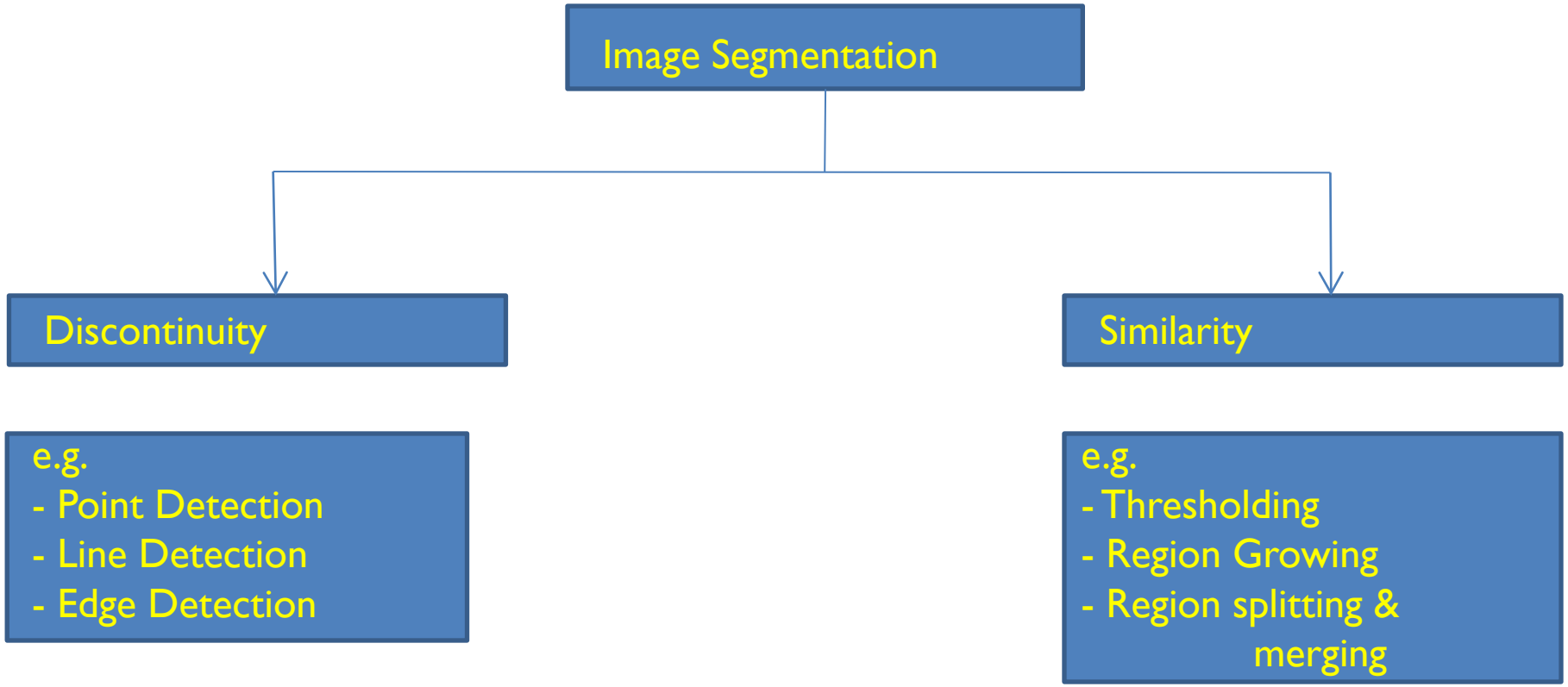


Segmentation methods

Image Segmentation:
- Thresholding techniques
- Clustering methods for segmentation
- Morphological operations

- Active contours (Snakes, Scissors, Level Sets)
- Split and merge (Watershed, Divisive & agglomerative clustering, Graph-based segmentation)
- K-means (parametric clustering)
- Mean shift (non-parametric clustering)
- Normalized cuts
- Graph cuts
- Graylevel thresholding

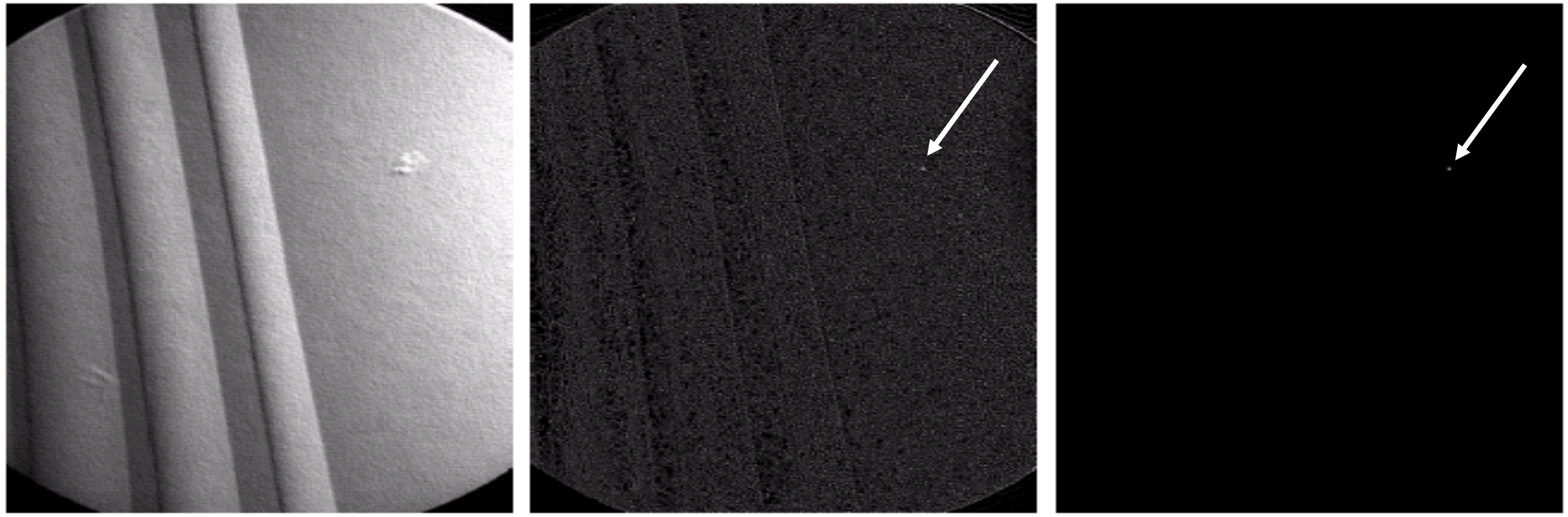
Classification of the Segmentation techniques



Point Detection

Apply detection mask, followed by threshold detection

−1	−1	−1
−1	8	−1
−1	−1	−1



a
b c d

FIGURE 10.2
(a) Point detection mask. (b) X-ray image of a turbine blade with a porosity. (c) Result of point detection. (d) Result of using Eq. (10.1-2). (Original image courtesy of X-TEK Systems Ltd.)

Point Detection

- It is based on Masking

-I	-I	-I
-I	8	-I
-I	-I	-I

- Find response R (response of convolution).
- The importance is strictly to detect points (isolated points).
- Compare and separate based on

$$|R| > T$$

T = non-negative threshold value

Point Detection(Example)

Original

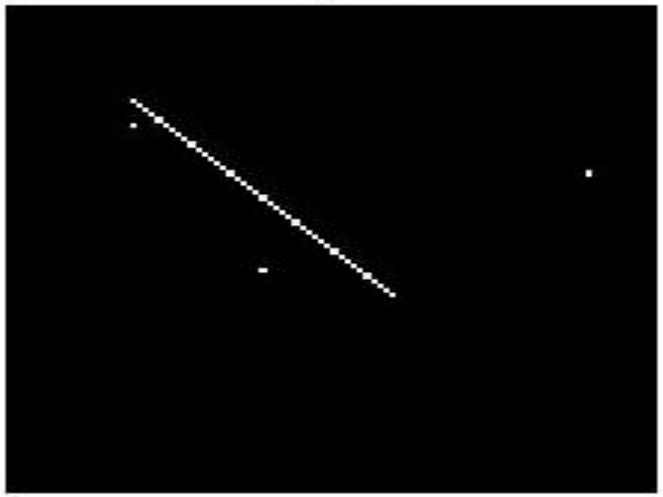
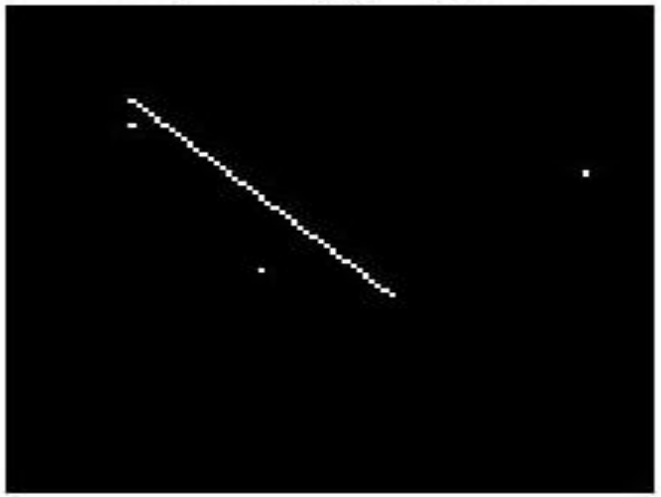


Image after applying mask



Thresholded image by T=8

Line Detection

−1	−1	−1
2	2	2
−1	−1	−1

Horizontal

−1	−1	2
−1	2	−1
2	−1	−1

+45°

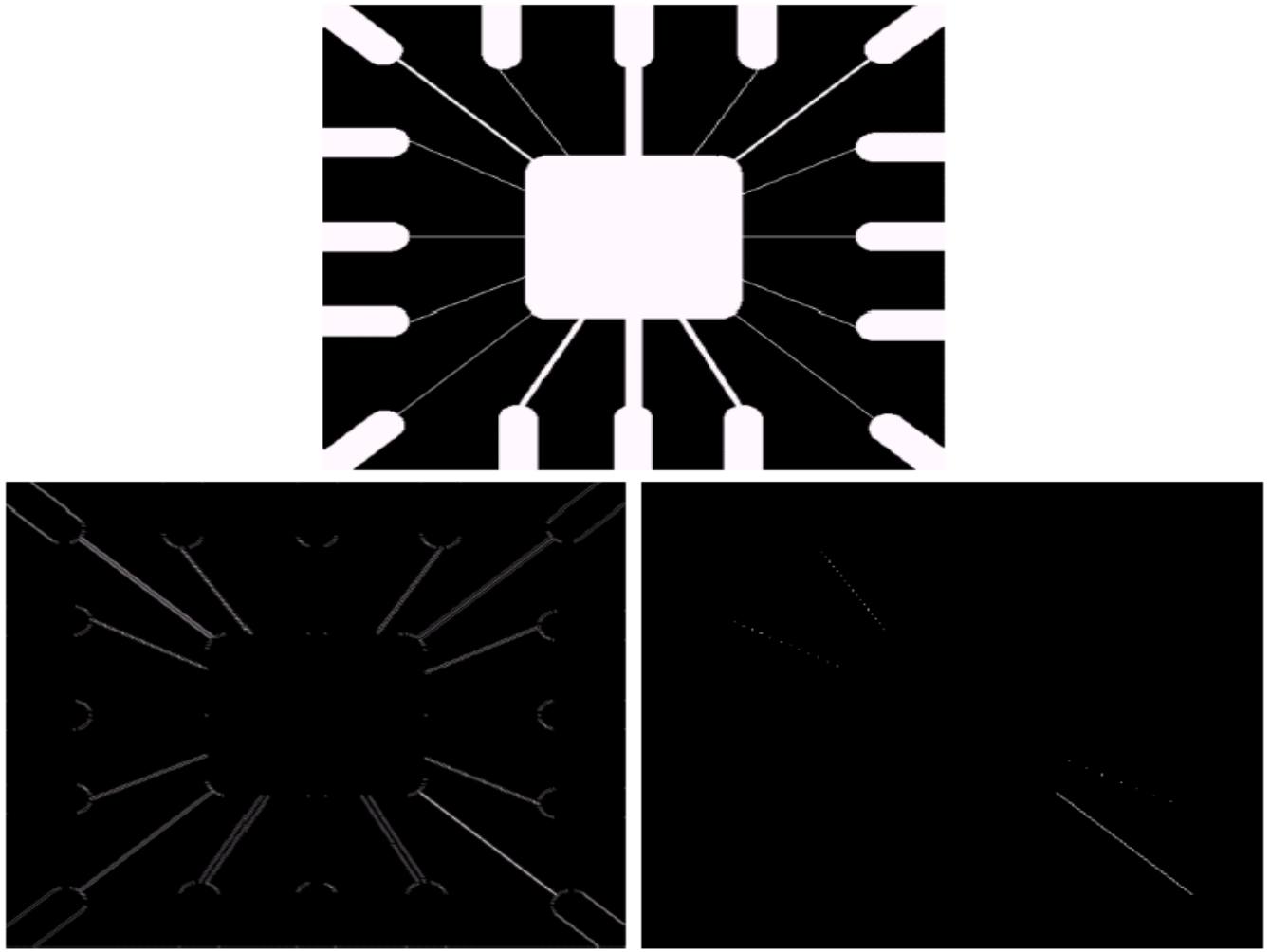
−1	2	−1
−1	2	−1
−1	2	−1

Vertical

2	−1	−1
−1	2	−1
−1	−1	2

−45°

Useful for detecting lines with width = 1.



a
b c

FIGURE 10.4 Illustration of line detection. (a) Binary wire-bond mask. (b) Absolute value of result after processing with −45° line detector. (c) Result of thresholding image (b).

Line Detection

Horizontal Line

-1	-1	-1
2	2	2
-1	-1	-1

45 degree inclined Line

-1	-1	2
-1	2	-1
2	-1	-1

Vertical Line

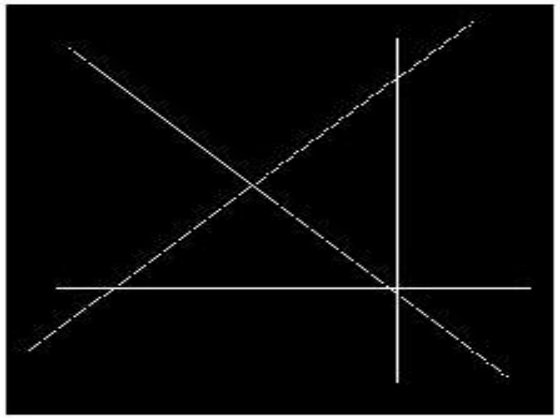
-1	2	-1
-1	2	-1
-1	2	-1

-45 degree inclined Line

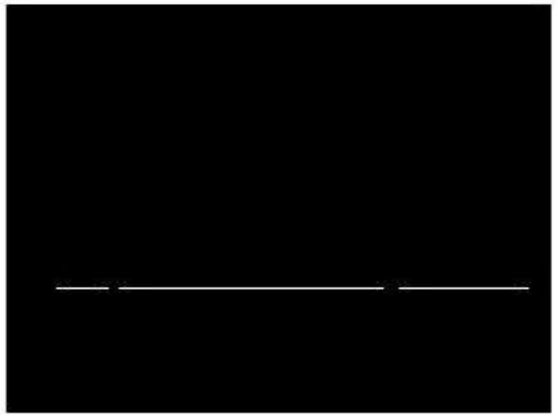
2	-1	-1
-1	2	-1
-1	-1	2

Line Detection(Cont.)

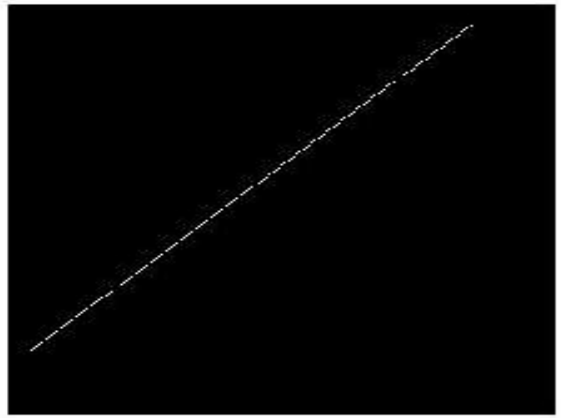
Original Image



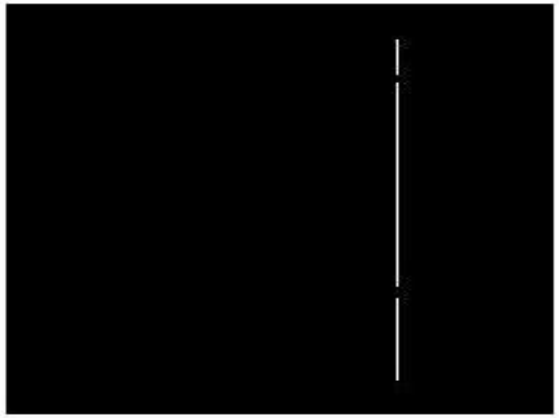
Horizontal line detection



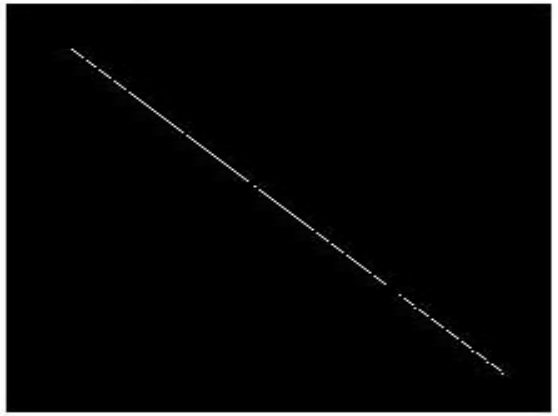
45 degree inclined line detection



Vertical line detection



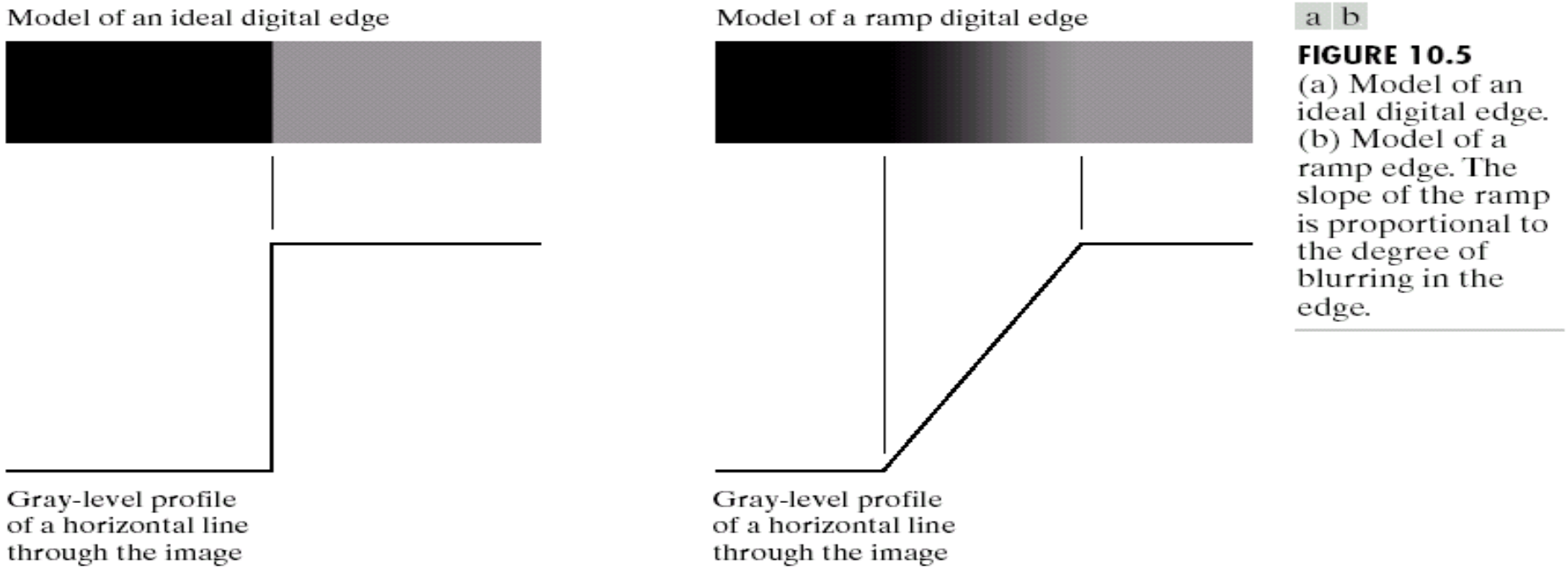
135 degree inclined line detection



Edge Detection

- Points and lines are special cases of edges.

- Edge detection is difficult since it is not clear what amounts to an edge!



a b

FIGURE 10.5
(a) Model of an ideal digital edge. (b) Model of a ramp edge. The slope of the ramp is proportional to the degree of blurring in the edge.

Edge Detection

Robert's Mask

-1	0
0	1

0	-1
1	0

Prewitt Operator

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

Sobel Operator

-1	-2	-1
0	0	0
1	2	1

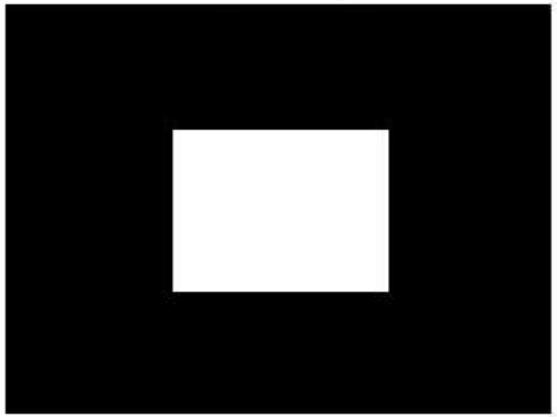
-1	0	1
-2	0	2
-1	0	1

Laplacian

-1	-1	-1
-1	8	-1
-1	-1	-1

Edge Detection(Example)

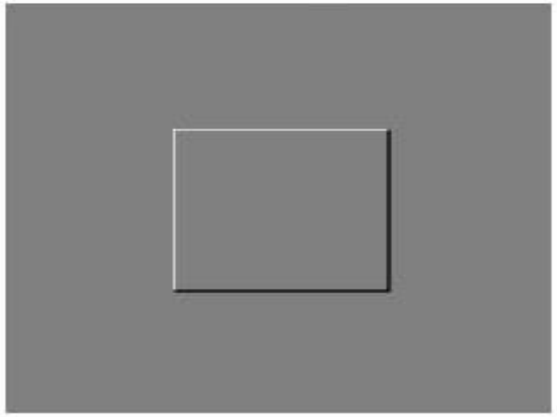
Original Image



Sobel-Horizontal Edge

Sobel-Vertical Edge

Sobel-Edge



Edge Detection(Example)

Original

Laplacian to unsharped image

After Thresholding by 15%

Combined Detection

- Multimask formulation makes possible development of a method to determine whether a pixel is most likely to be an isolated point or part of a line or an edge.

Edge Linking and Boundary Detection

- Intensity discontinuity can be utilized to find boundary.

- The lagging part of boundary detection using intensity discontinuity is that the boundary may not be completely defined because of
    - Noise
    - Breaks in boundary due to non-uniform illumination

- So, after edge detection, edge linking process is carried out to assemble edge pixels into meaningful boundary


DL57 DEU

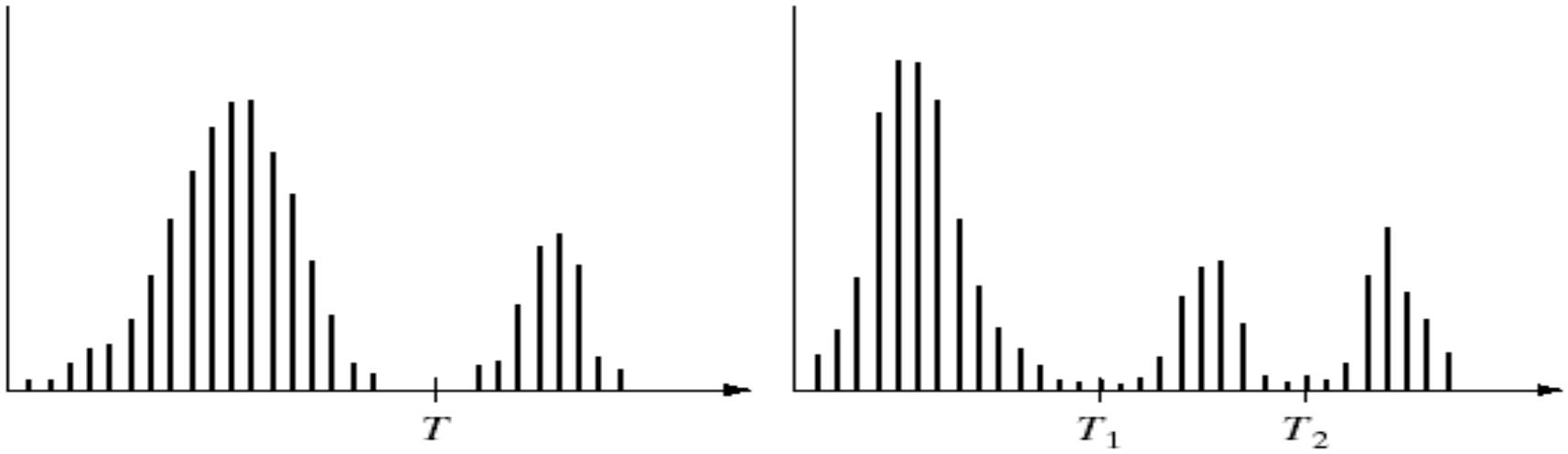

DL57 DEU

Boundary Extraction

- Edge detection classifies individual pixels to be on an edge or not.
- Isolated edge pixels is more likely to be noise rather than a true edge.
- Adjacent or connected edge pixels should be linked together to form boundary of regions that segment the image.

- Edge linking methods:
  - Local processing
  - Hough transform
  - Graphic theoretic method
  - Dynamic programming

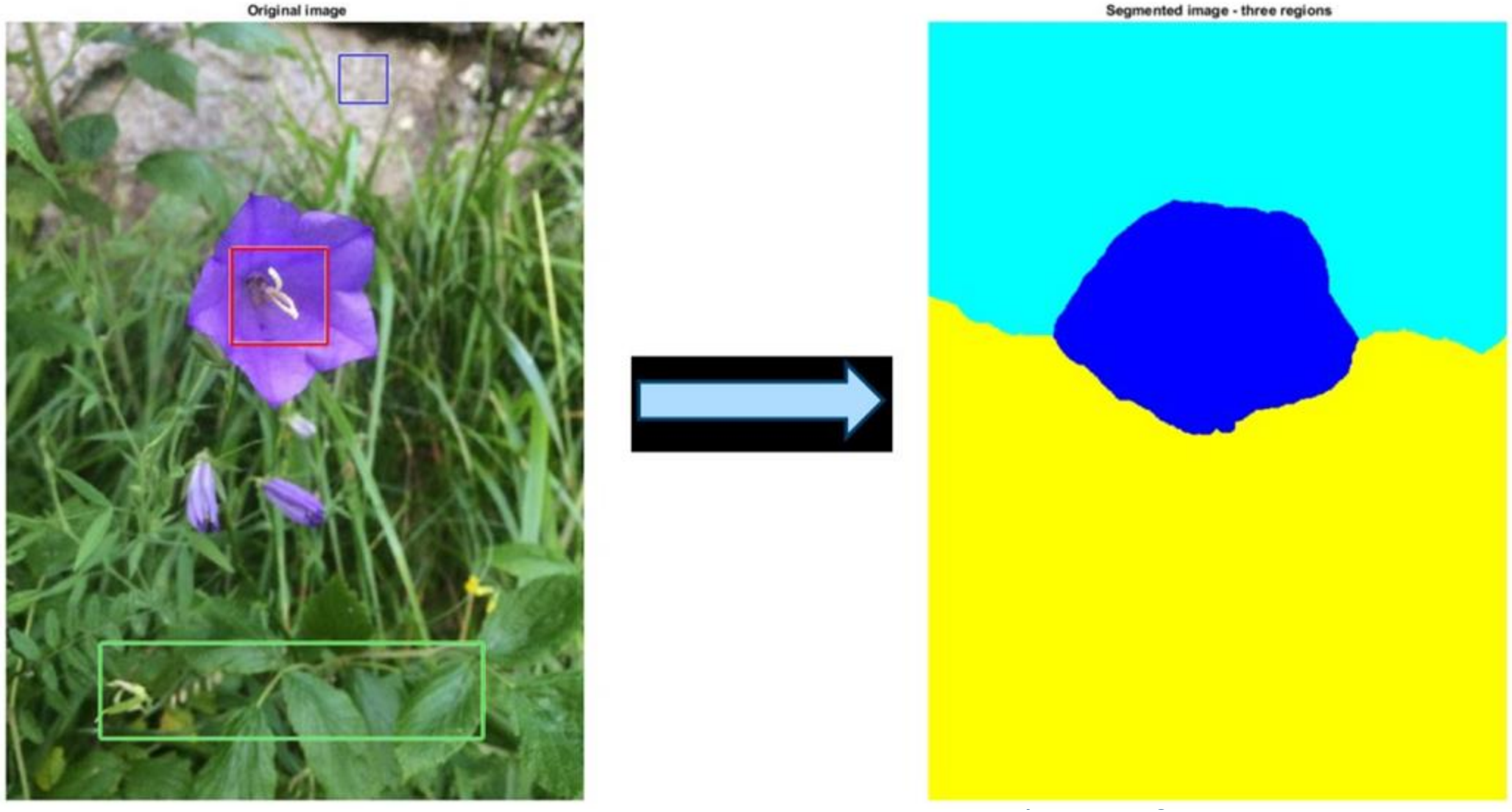
Threshold Segmentation



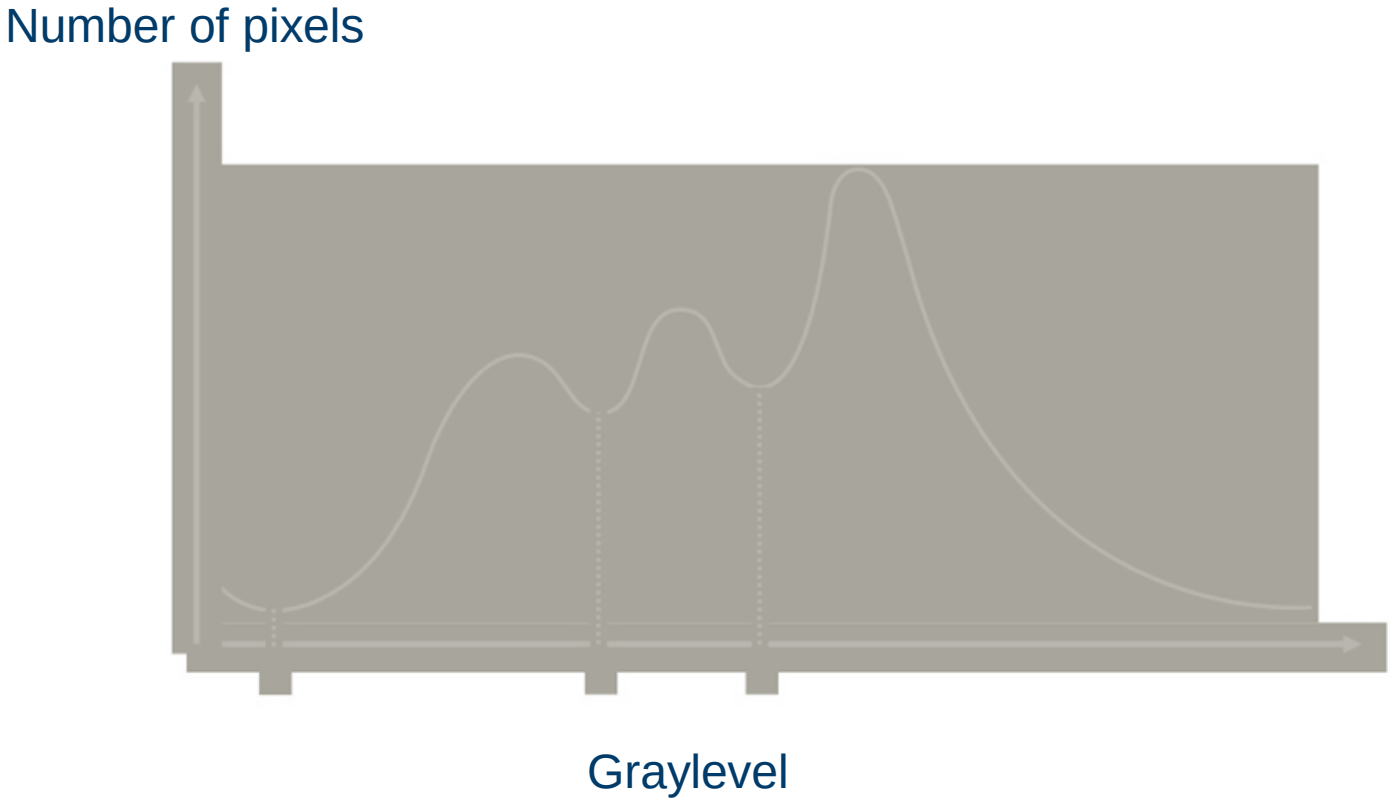
a b

FIGURE 10.26 (a) Gray-level histograms that can be partitioned by (a) a single threshold, and (b) multiple thresholds.

Colour Segmentation - Example



Segmentation by thresholding



Otsu's method:

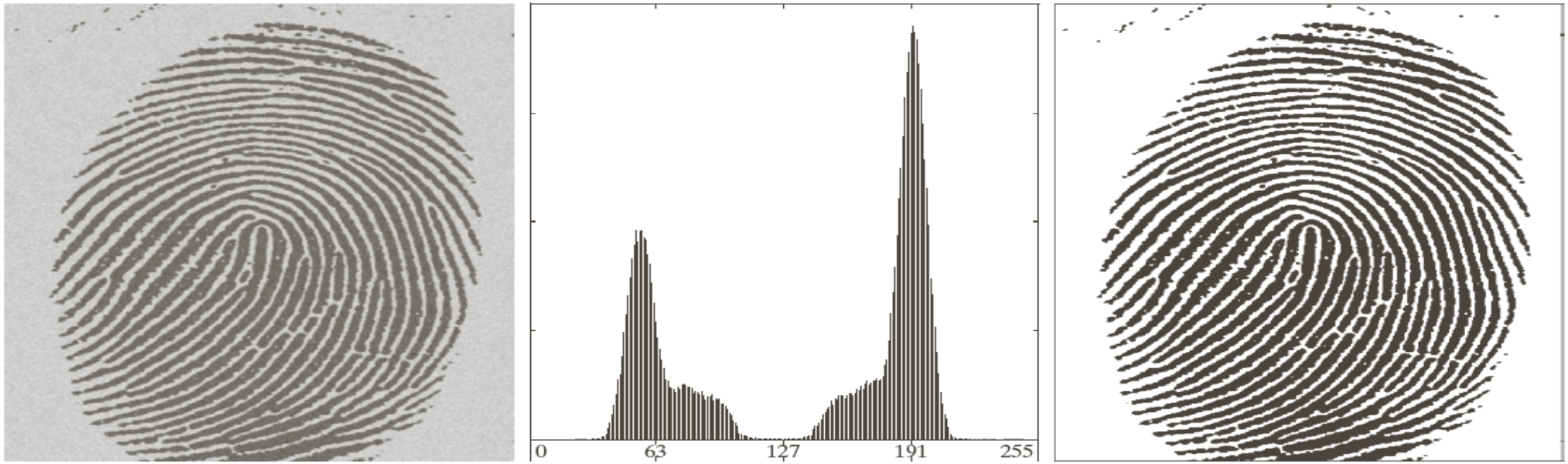
- Automatic clustering-based thresholding
- Minimization of intra-class variance
- Analog to Fisher's Discriminant Analysis

Basic Global Thresholding

1. Select an initial estimate for the global threshold, T.
2. Segment the image using T. It will produce two groups of pixels: G1 consisting of all pixels with intensity values > T and G2 consisting of pixels with values $\leq$T.
3. Compute the average intensity values m1 and m2 for the pixels in G1 and G2, respectively.
4. Compute a new threshold value.

$$T = \frac{1}{2}(m1 + m2)$$

5. Repeat Steps 2 through 4 until the difference between values of T in successive iterations is smaller than a predefined parameter ΔT



a b c

FIGURE 10.38 (a) Noisy fingerprint. (b) Histogram. (c) Segmented result using a global threshold (the border was added for clarity). (Original courtesy of the National Institute of Standards and Technology.)

Optimum Global Thresholding Using Otsu's Method

- Principle: maximizing the between-class variance

Let $\{0, 1, 2, \ldots, L-1\}$ denote the L distinct intensity levels in a digital image of size $M \times N$ pixels, and let n_i denote the number of pixels with intensity i.

$$p_i = n_i / MN \quad \text{and} \quad \sum_{i=0}^{L-1} p_i = 1$$

k is a threshold value, $C_1 \rightarrow [0, k]$, $C_2 \rightarrow [k+1, L-1]$

$$P_1(k) = \sum_{i=0}^{k} p_i \quad \text{and} \quad P_2(k) = \sum_{i=k+1}^{L-1} p_i = 1 - P_1(k)$$

Optimum Global Thresholding Using Otsu's Method

The mean intensity value of the pixels assigned to class C_1 is

$$m_1(k) = \sum_{i=0}^{k} iP(i / C_1) = \frac{1}{P_1(k)} \sum_{i=0}^{k} ip_i$$

The mean intensity value of the pixels assigned to class C_2 is

$$m_2(k) = \sum_{i=k+1}^{L-1} iP(i / C_2) = \frac{1}{P_2(k)} \sum_{i=k+1}^{L-1} ip_i$$

$$P_1 m_1 + P_2 m_2 = m_G \quad \text{(Global mean value)}$$

Optimum Global Thresholding Using Otsu's Method

Between-class variance, σ_B^2 is defined as

$$\sigma_B^2 = P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2$$

$$= P_1 P_2 (m_1 - m_2)^2$$

$$= \frac{\left[m_G P_1 - m_1 P_1\right]^2}{P_1(1-P_1)}$$

$$= \frac{\left[m_G P_1 - m\right]^2}{P_1(1-P_1)}$$

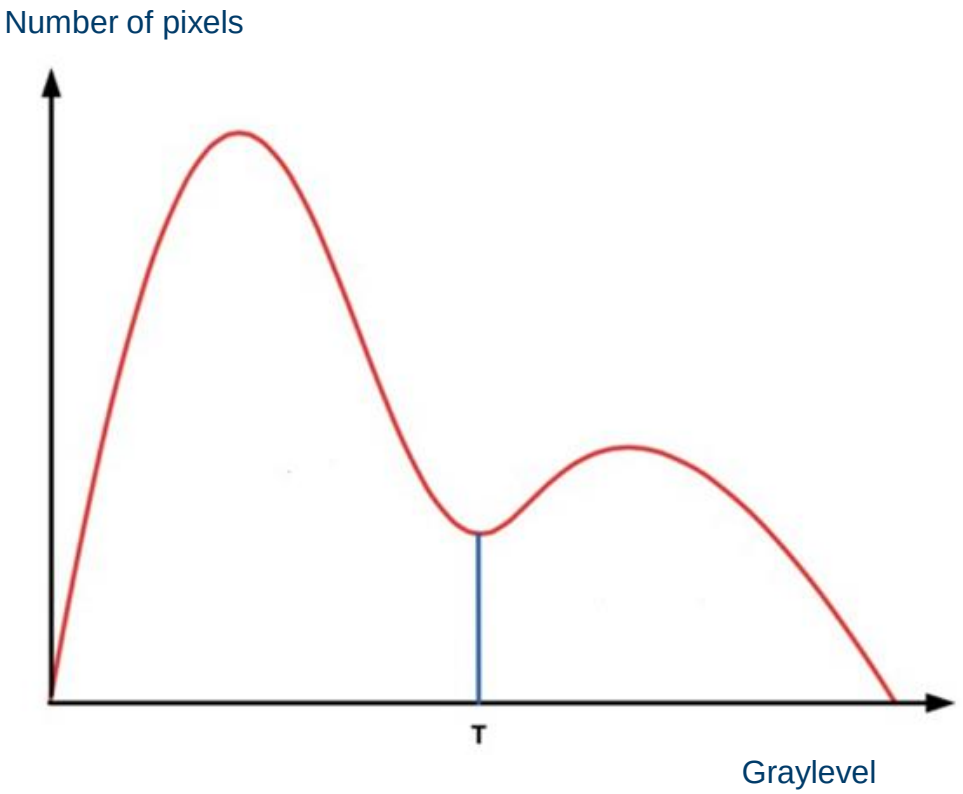
Otsu's Algorithm: Summary

1. Compute the normalized histogram of the input image. Denote the components of the histogram by p_i, i=0, 1, …, L-1.
2. Compute the cumulative sums, $P_1(k)$, for k = 0, 1, …, L-1.
3. Compute the cumulative means, m(k), for k = 0, 1, …, L-1.
4. Compute the global intensity mean, m_G.
5. Compute the between-class variance, for k = 0, 1, …, L-1.
6. Obtain the Otsu's threshold, k*.
7. Obtain the separability measure.

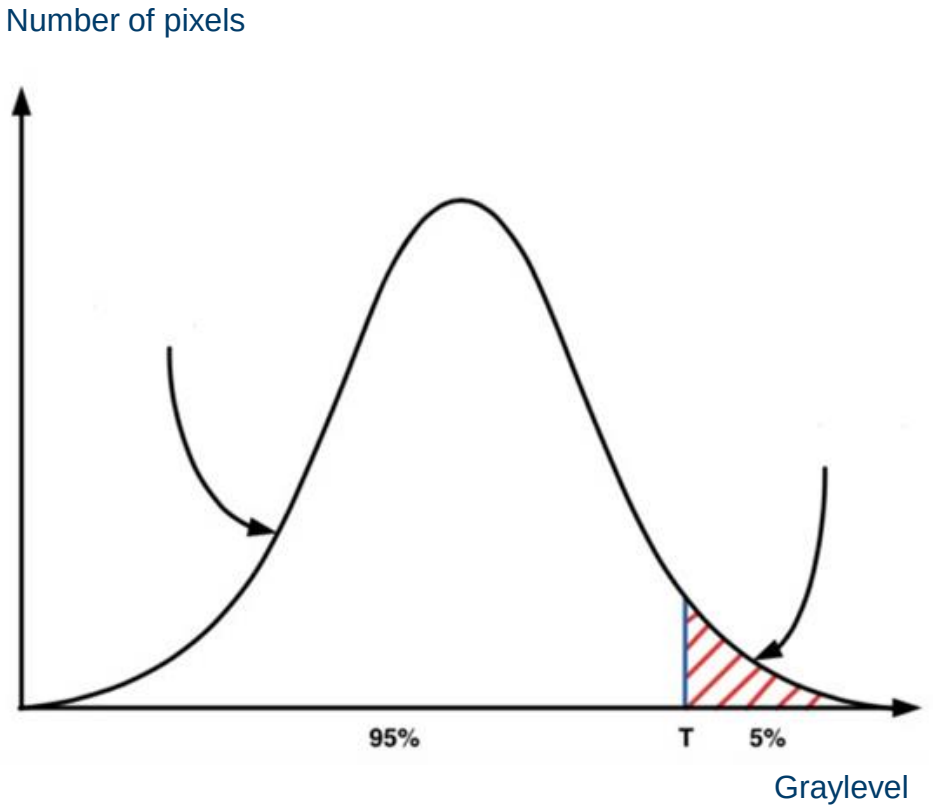
Thresholding with Otsu’s method

3 thresholds with 4 classes

Binary segmentation - foreground vs. background



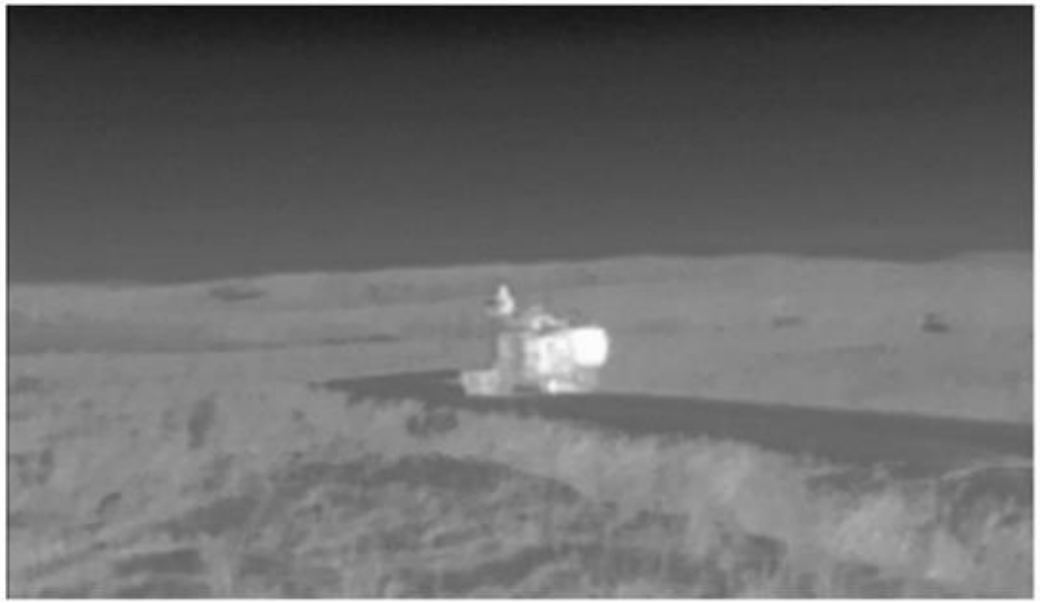
Threshold between two populations



Foreground

Threshold at given percentile

Binary thresholding - Object detection



Thermal image

Thresholded image (Otsu's method)

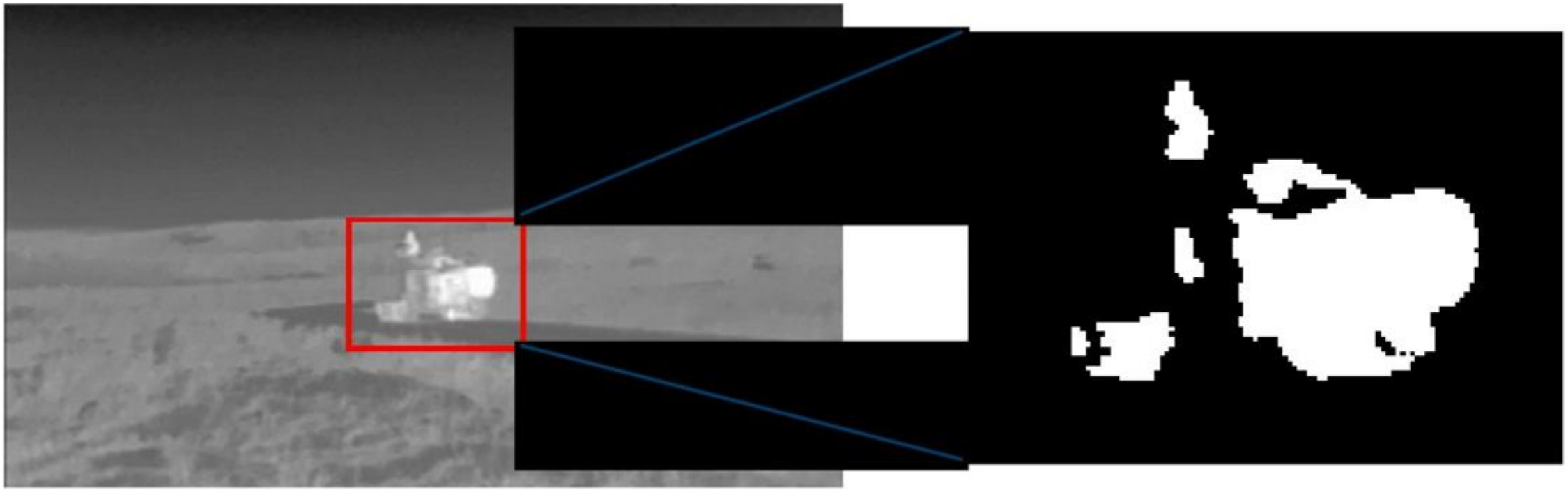
Global threshold selection · threshold *too low* for detection of the object of interest

Manual thresholding

Medium threshold

High threshold

Local thresholding



Threshold computed from graylevel statistics in selected window (Otsu’s method)

Local thresholding using edge information

Edge image (Canny edge detector applied to selected window)

Threshold = average graylevel along edges

Thresholded window

Object detection in video sequences (visible light)

Daylight video frame

Thresholded difference image

- Change detection
- Absolute difference image (Current image - time averaged background image)
- Thresholding of difference image, i.e. Otsu's method
- Requires fixed camera (or registration of images)

Region-Based Segmentation

Region Growing

1. Region growing is a procedure that groups pixels or subregions into larger regions.

2. The simplest of these approaches is ***pixel aggregation***, which starts with a set of "seed" points and from these grows regions by appending to each seed points those **neighboring pixels** that have **similar properties** (such as gray level, texture, color, shape).

3. Region growing based techniques are better than the edge-based techniques in noisy images where edges are difficult to detect.

Example: Region Growing based on 8-connectivity

$f(x, y)$: input image array
$S(x, y)$: seed array containing 1s (seeds) and 0s
$Q(x, y)$: predicate

Segmentation by clustering

- Cluster together (pixels, tokens, etc.) that belong together
- Agglomerative clustering
  - attach closest to cluster it is closest to
  - repeat
- Divisive clustering
  - split cluster along best boundary
  - repeat

- Point-Cluster distance
  - single-link clustering
  - complete-link clustering
  - group-average clustering
- Dendrograms
  - yield a picture of output as clustering process continues

Clustering Problem Statement

- Using centroid to represent the huge numbers of clusters
- Hierarchical clustering, we can change the number of cluster anytime during process if we want.
- Partitional clustering , we have to decide the number of clustering we need first before we begin the process.

- Given a set of vectors $\{x_k; 1 \leq k \leq K\}$, find a set of M clustering centers $\{\mathbf{w}(i); 1 \leq i \leq c\}$ such that each x_k is assigned to a cluster, say, $\mathbf{w}(i^*)$, according to a distance (distortion, similarity) measure $d(x_k, \mathbf{w}(i))$ such that the average distortion

$$D = \frac{1}{K} \sum_{i=1}^{c} \sum_{k=1}^{K} I(x_k, i) d(x_k, W(i))$$

  is minimized.
- $\mathbf{I}(x_k,i)$ = 1 if x is assigned to cluster i with cluster center w(I); and = 0 otherwise -- indicator function.

Hierarchical clustering

Algorithm of hierarchical agglomeration(built)：

A. See every single data as a cluster C_i.

B. Find out C_i , C_j for the distance is the shortest.

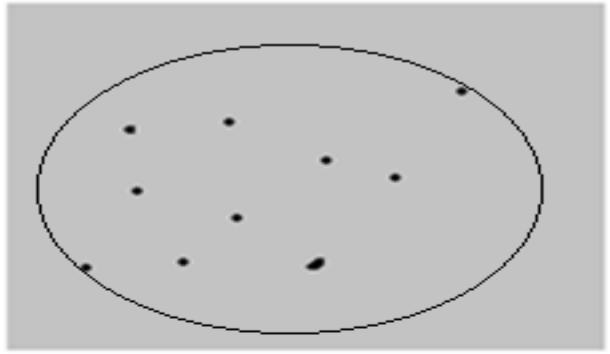
C. Repeat the steps until satisfies our demand.

$d(a,b)$ as the distance between data a and b

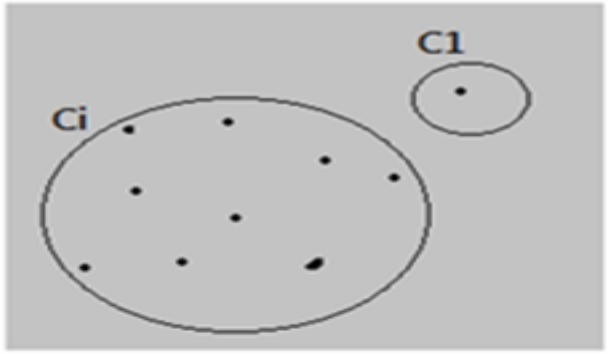
Algorithm of hierarchical division (break up)：

- Diameter of cluster

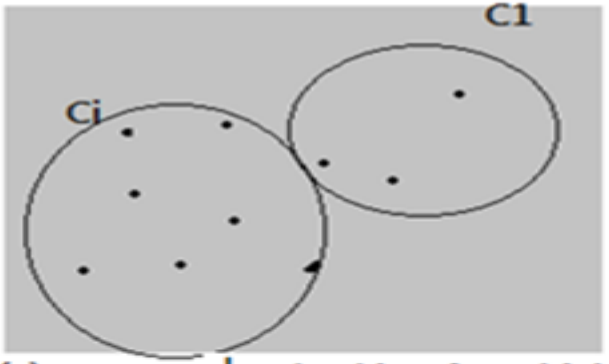
$$D(C_i) = \max(d(\mathrm{a},\mathrm{b}))\ ,\ for\ \forall \mathrm{a} \in C_i\ ,\ \forall \mathrm{b} \in C_i$$



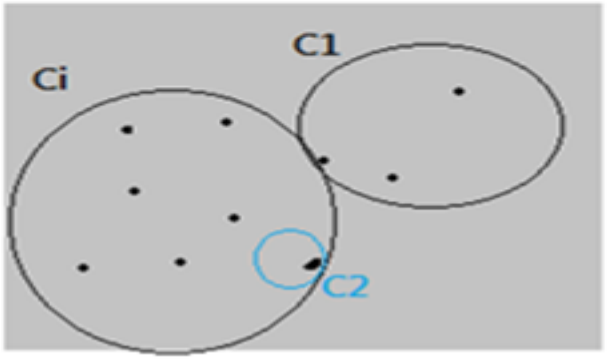
(a) original image as a cluster C



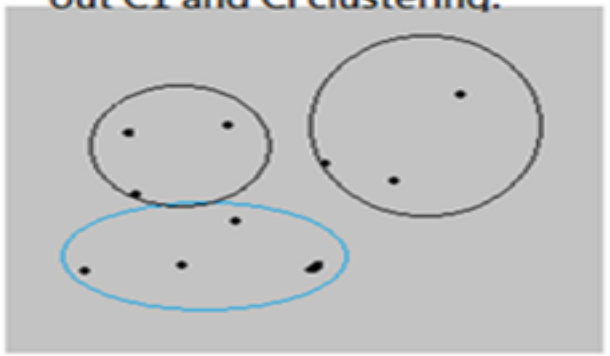
(b)use step b and step c to find out C1 and Ci clustering.



(c)use step d to looking for which point is closer to C1 than Ci



(d) Find out the biggest diameter and to repeat stpet b to step d



(e) The end of process

1) See the whole database as one cluster.

2) Find out the cluster having the biggest diameter

3) $\max\left(d(x,c)\right)$, for $\forall x \in C$.

4) Split out x as a new cluster C_1, and see the rest data points as C_i.

5) If $d(y,C_i) > d(y,C_1)$, for $\forall y \in C_i$ then split y out of C_i and classify it to C_1

6) Back to step2 and continue the algorithm until C_1 and C_i is not change anymore.

Simple clustering algorithms

Algorithm 15.3: Agglomerative clustering, or clustering by merging

```
Make each point a separate cluster
Until the clustering is satisfactory
     Merge the two clusters with the
       smallest inter-cluster distance
end
```

Algorithm 15.4: Divisive clustering, or clustering by splitting

```
Construct a single cluster containing all points
Until the clustering is satisfactory
     Split the cluster that yields the two
       components with the largest inter-cluster distance
end
```

K-Means

- Choose a fixed number of clusters
- Choose cluster centers and point-cluster allocations to minimize error
- can’t do this by search, because there are too many possible allocations.

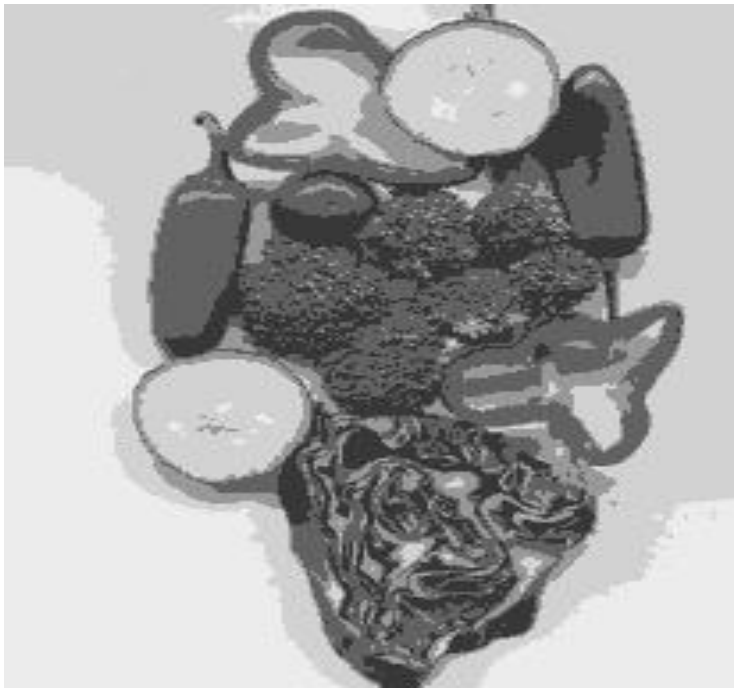
- Algorithm
  - fix cluster centers; allocate points to closest cluster
  - fix allocation; compute best cluster centers
- x could be any set of features for which we can compute a distance (careful about scaling)

Image Segmentation by K-Means

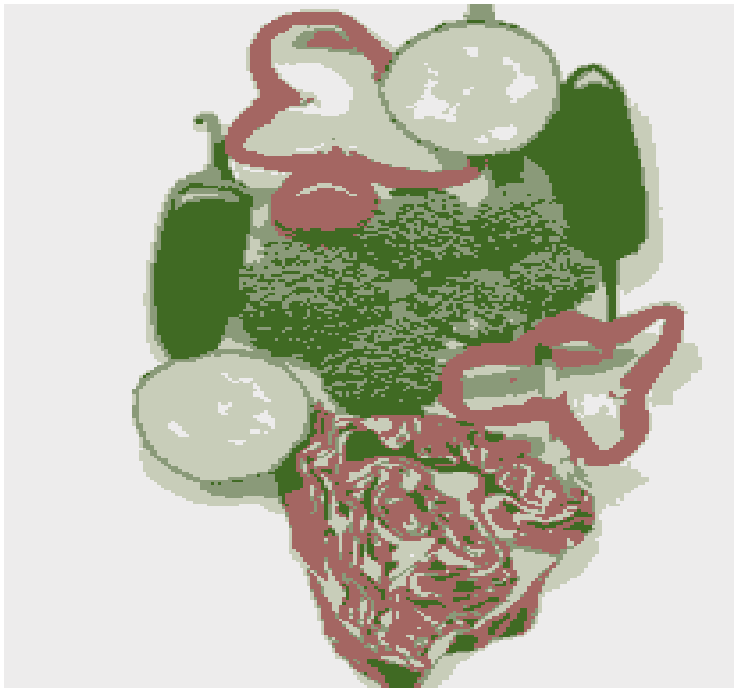
- Select a value of K
- Select a feature vector for every pixel (color, texture, position, or combination of these etc.)
- Define a similarity measure between feature vectors (Usually Euclidean Distance).
- Apply K-Means Algorithm.
- Apply Connected Components Algorithm.
- Merge any components of size less than some threshold to an adjacent component that is most similar to it.

K-Means Clustering

Image



Clusters on intensity



Clusters on color

K-means clustering using intensity alone and color alone

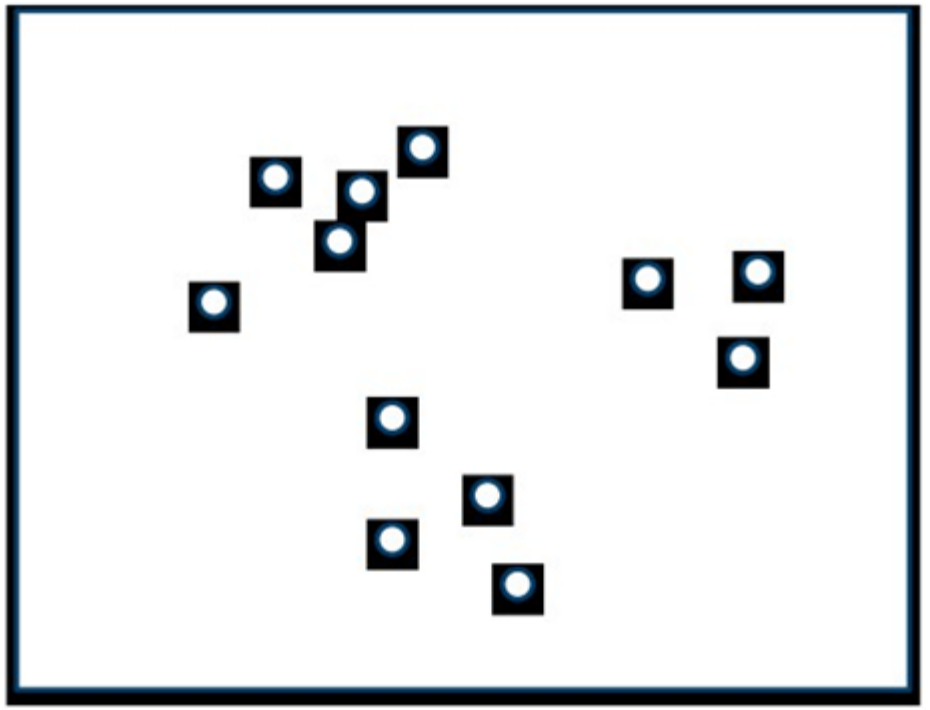
K-means (parametric) clustering

1. Select K points (for example randomly) as initial cluster centers

2. Assign each sample to nearest cluster center

3. Compute new cluster centers (i.e. sample means)

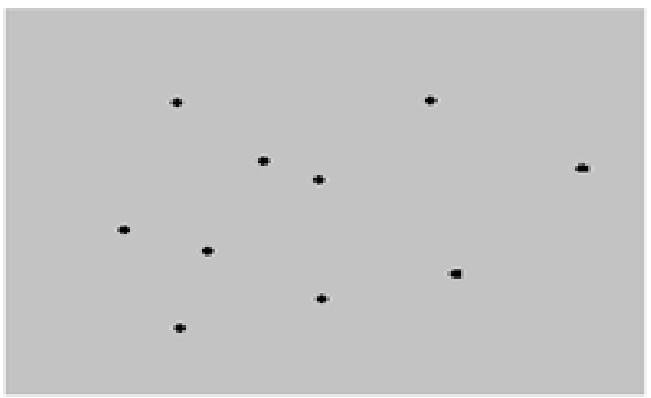
4. Repeat steps 2 and 3 until no further re-assignments are possible.

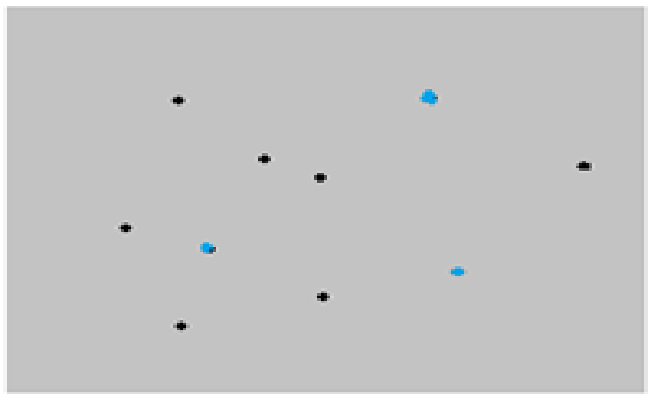


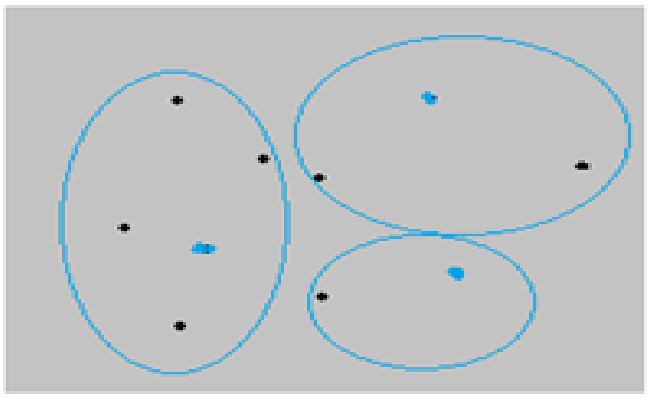
Unlabeled dataset

Partitional clustering

- Decide the numbers of the cluster(k-means)

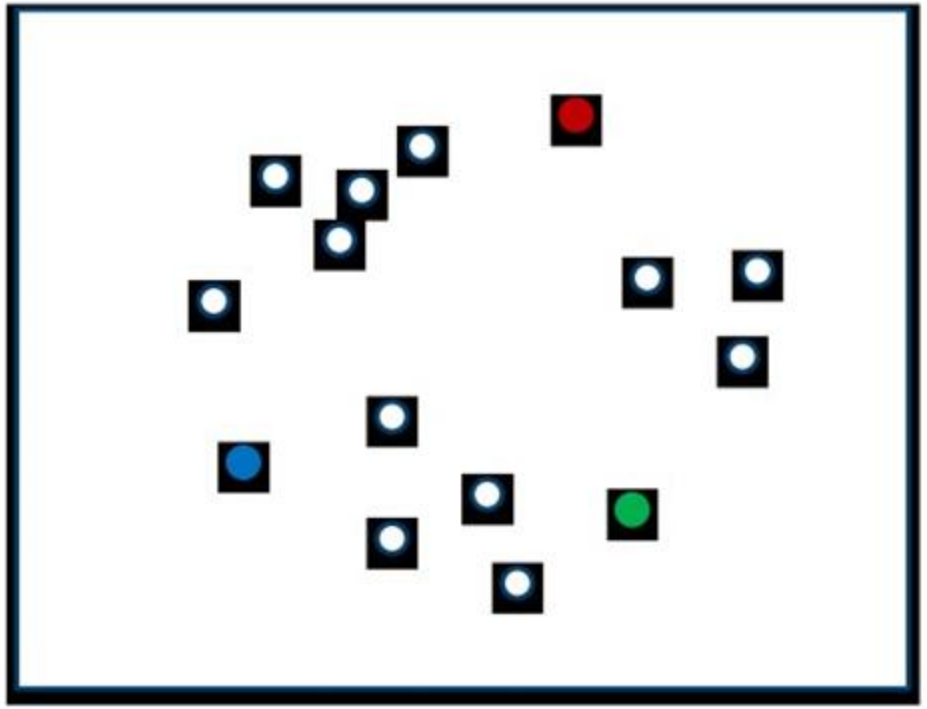




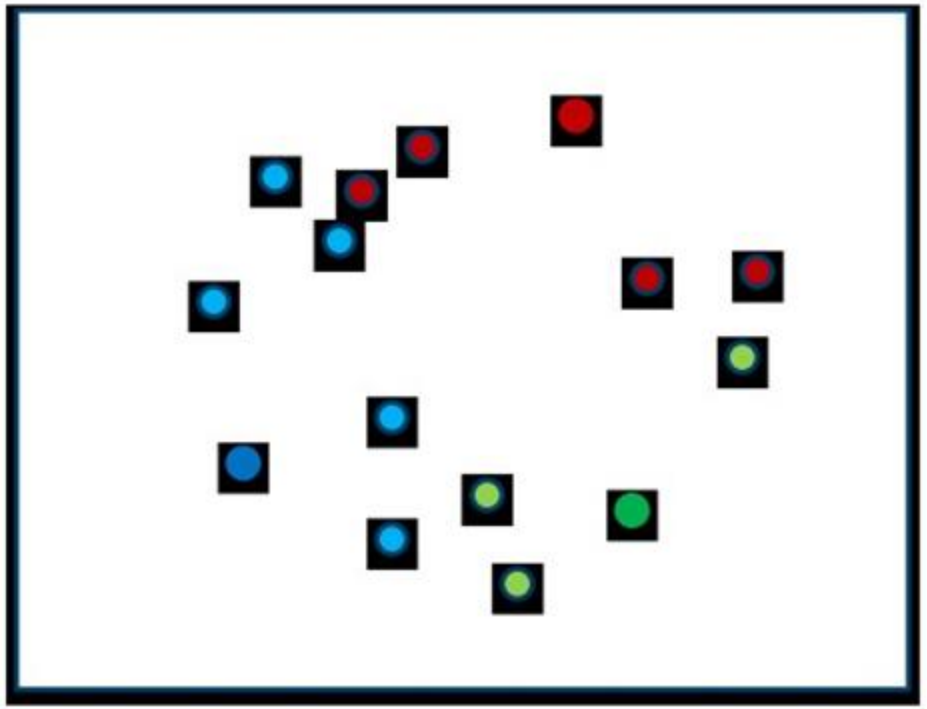


- Problem:
- Initial problem

K-means clustering

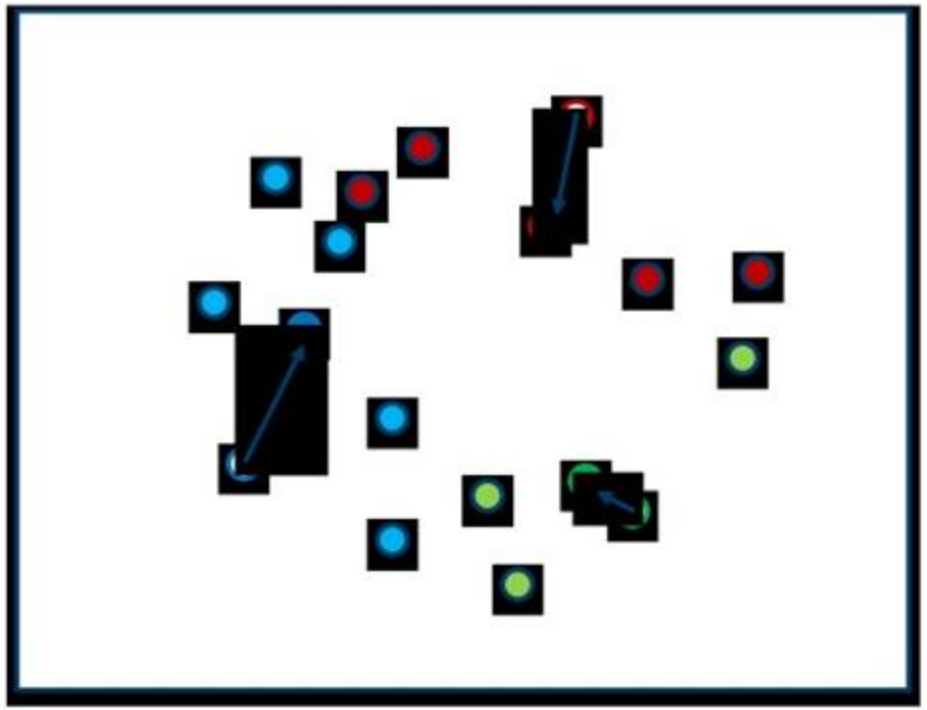


Initial cluster centers (red, green and blue points)

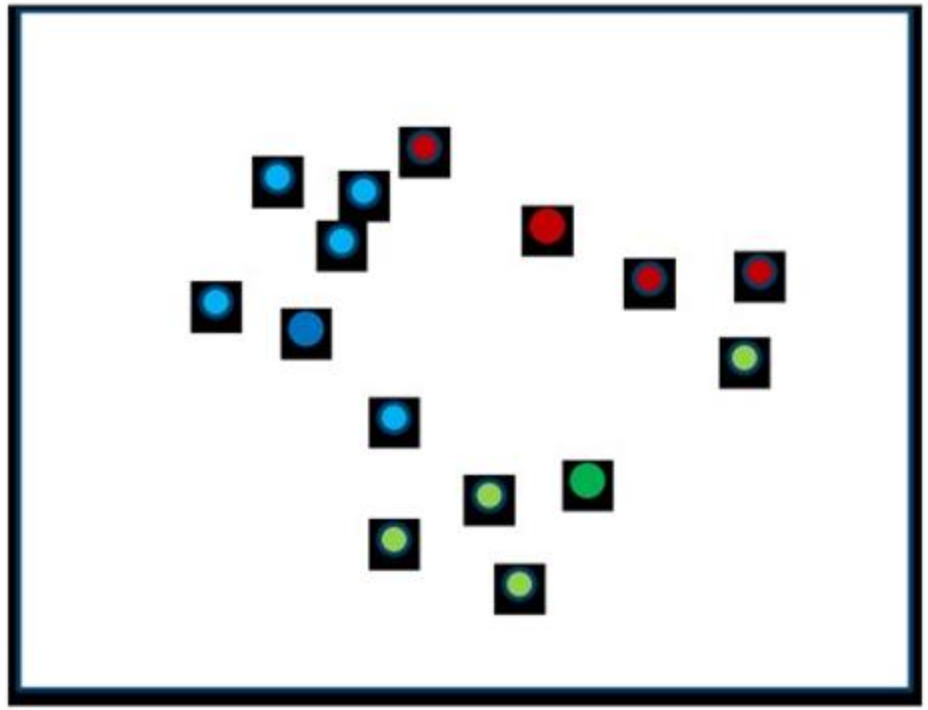


Samples assigned to nearest cluster center

K-means clustering

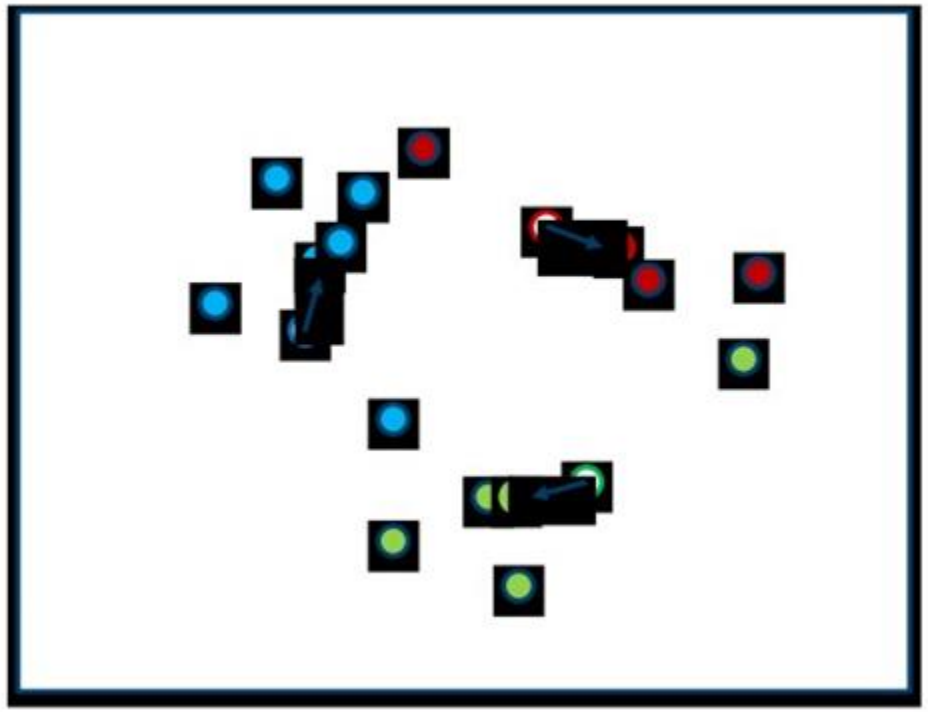


Re-computed cluster centers

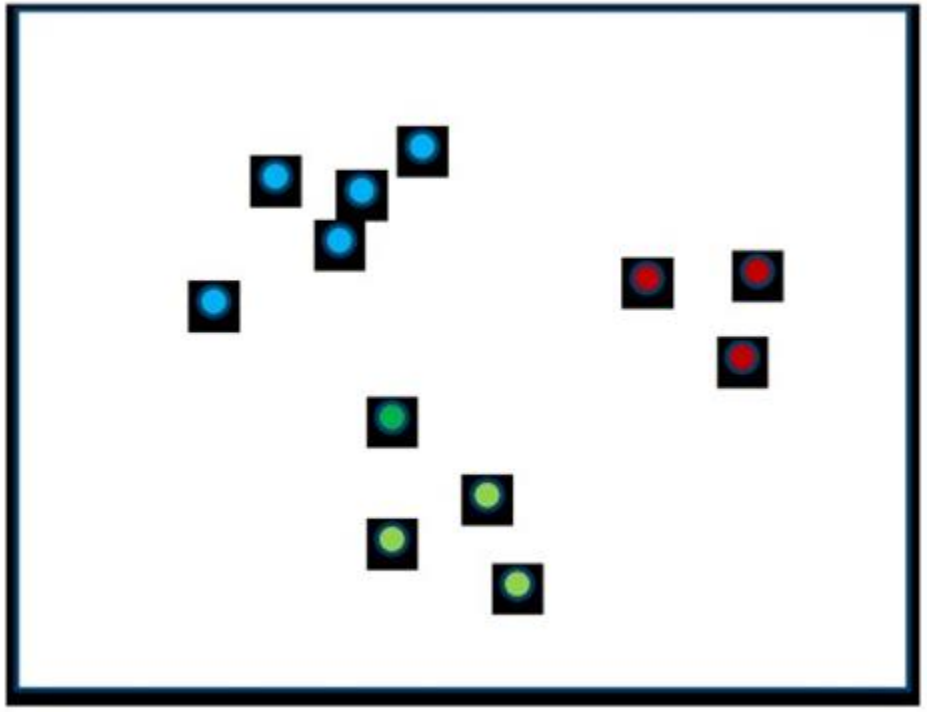


Samples re-assigned to new cluster centers

K-means clustering



Re-computed cluster centers



Final clustering

K-means clustering using color

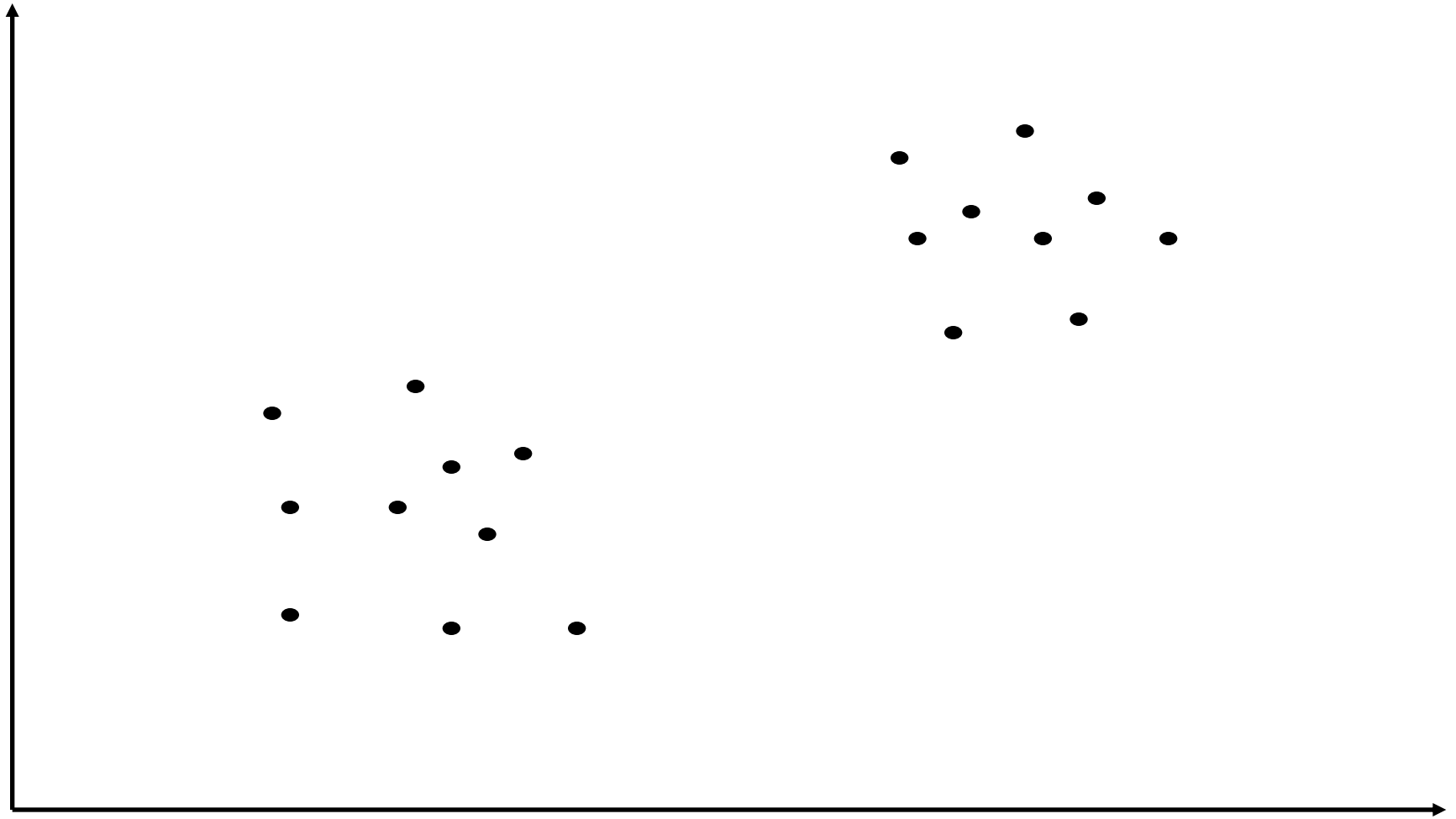
Original image

Clustered image - 10 clusters

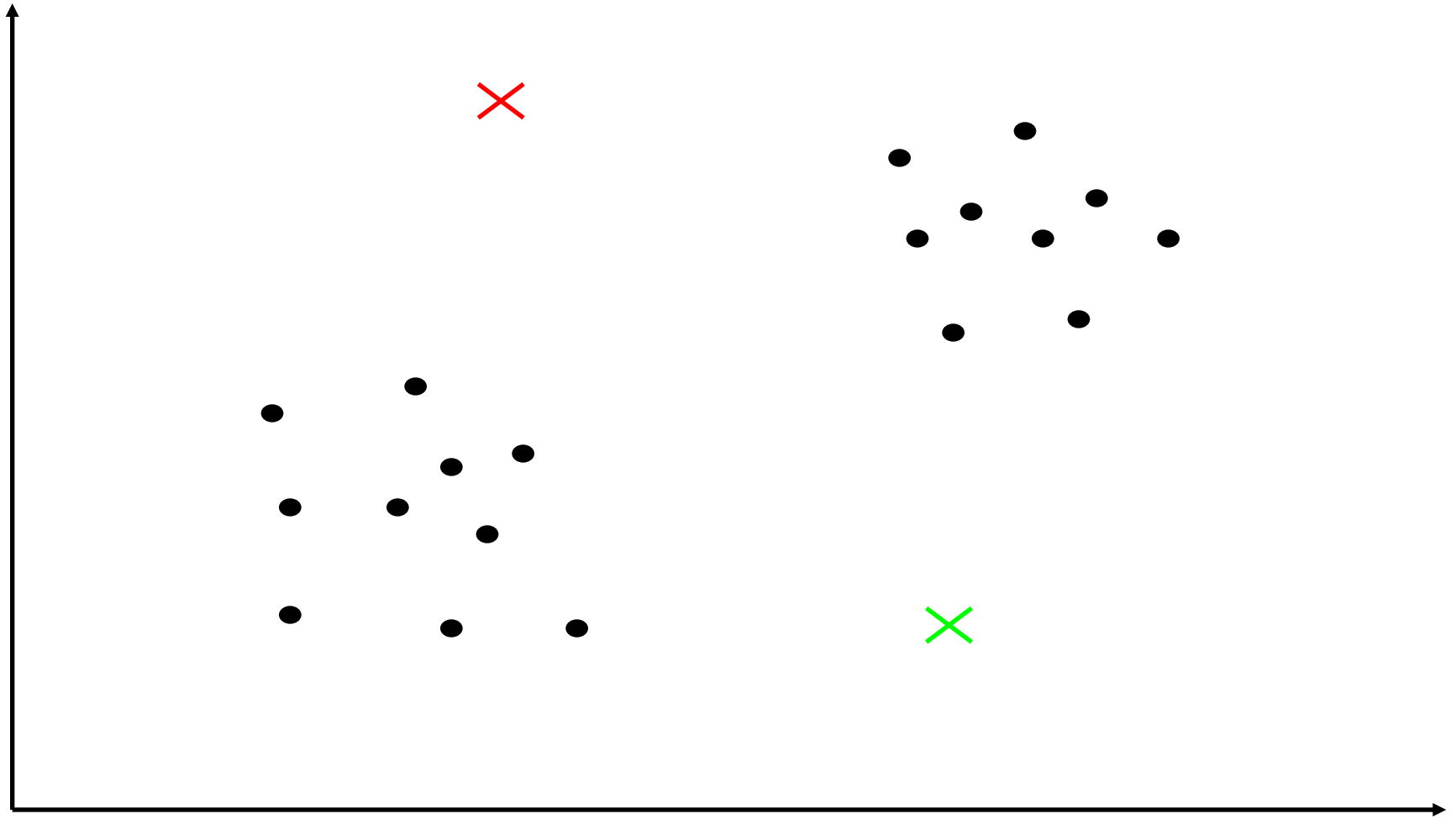
K-means Clustering

- Partition the data points into K clusters randomly. Find the centroids of each cluster.
- For each data point:
  - Calculate the distance from the data point to each cluster.
  - Assign the data point to the closest cluster.
- Recompute the centroid of each cluster.
- Repeat steps 2 and 3 until there is no further change in the assignment of data points (or in the centroids).

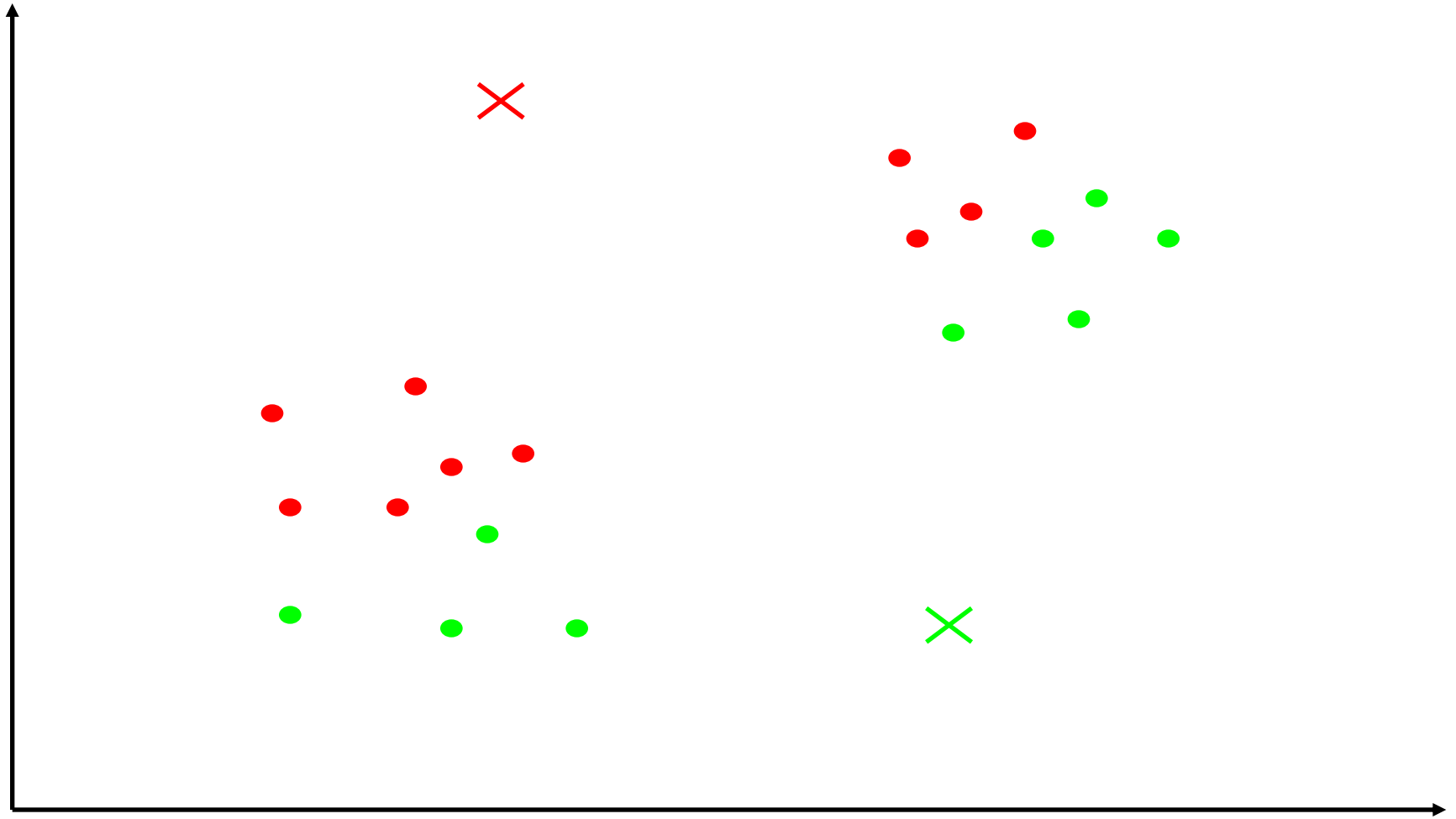
K-Means Clustering



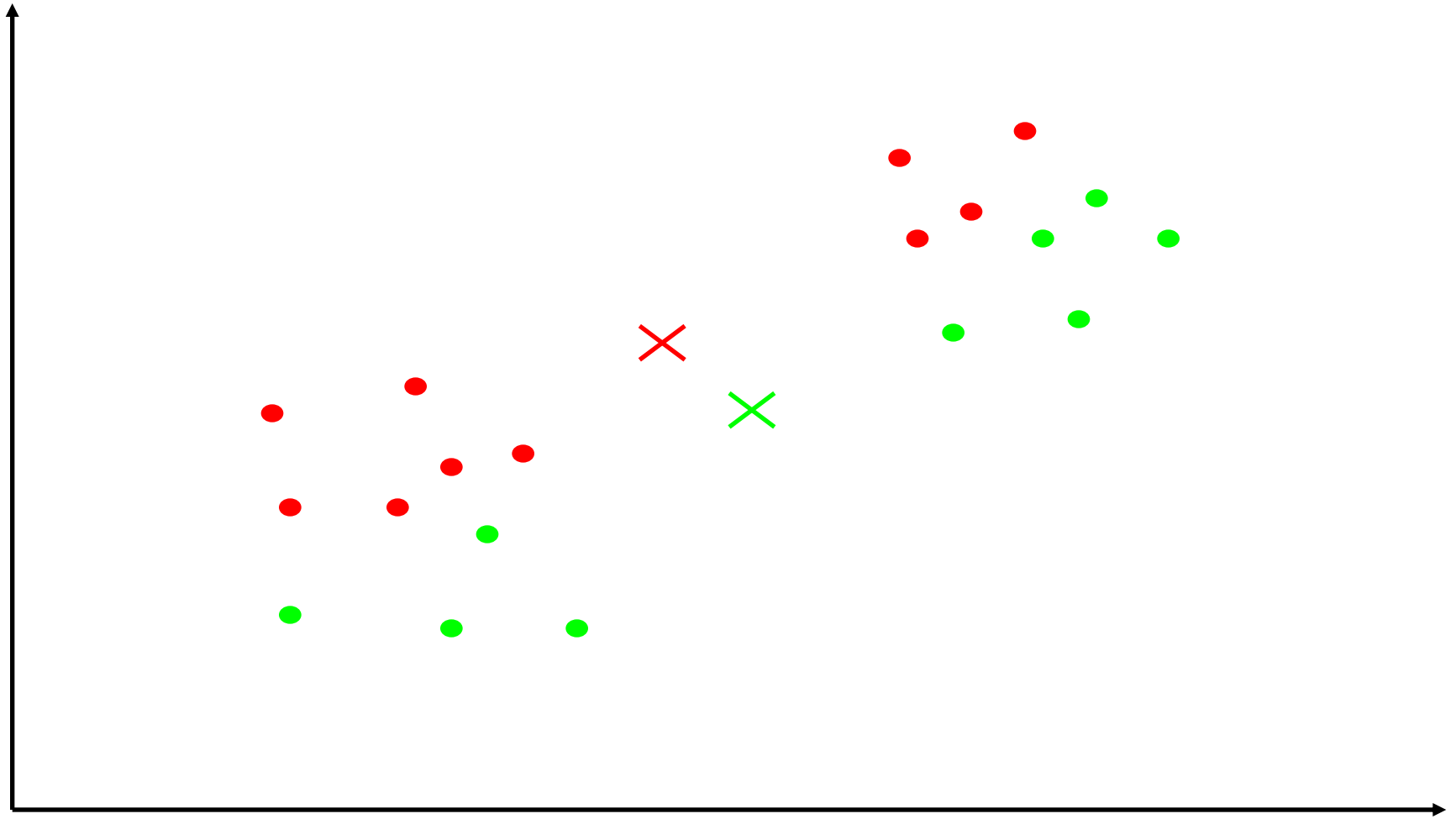
K-Means Clustering



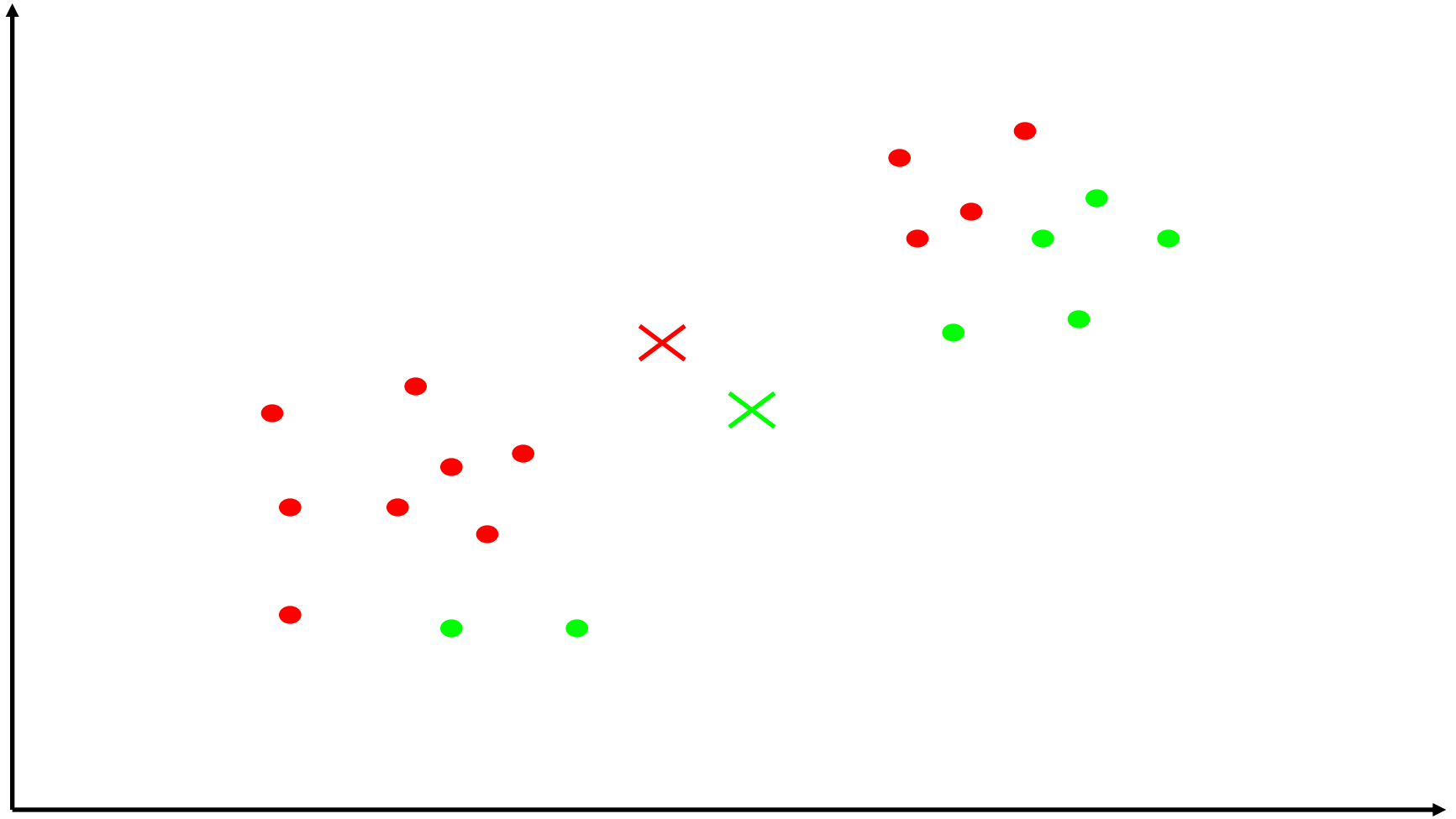
K-Means Clustering



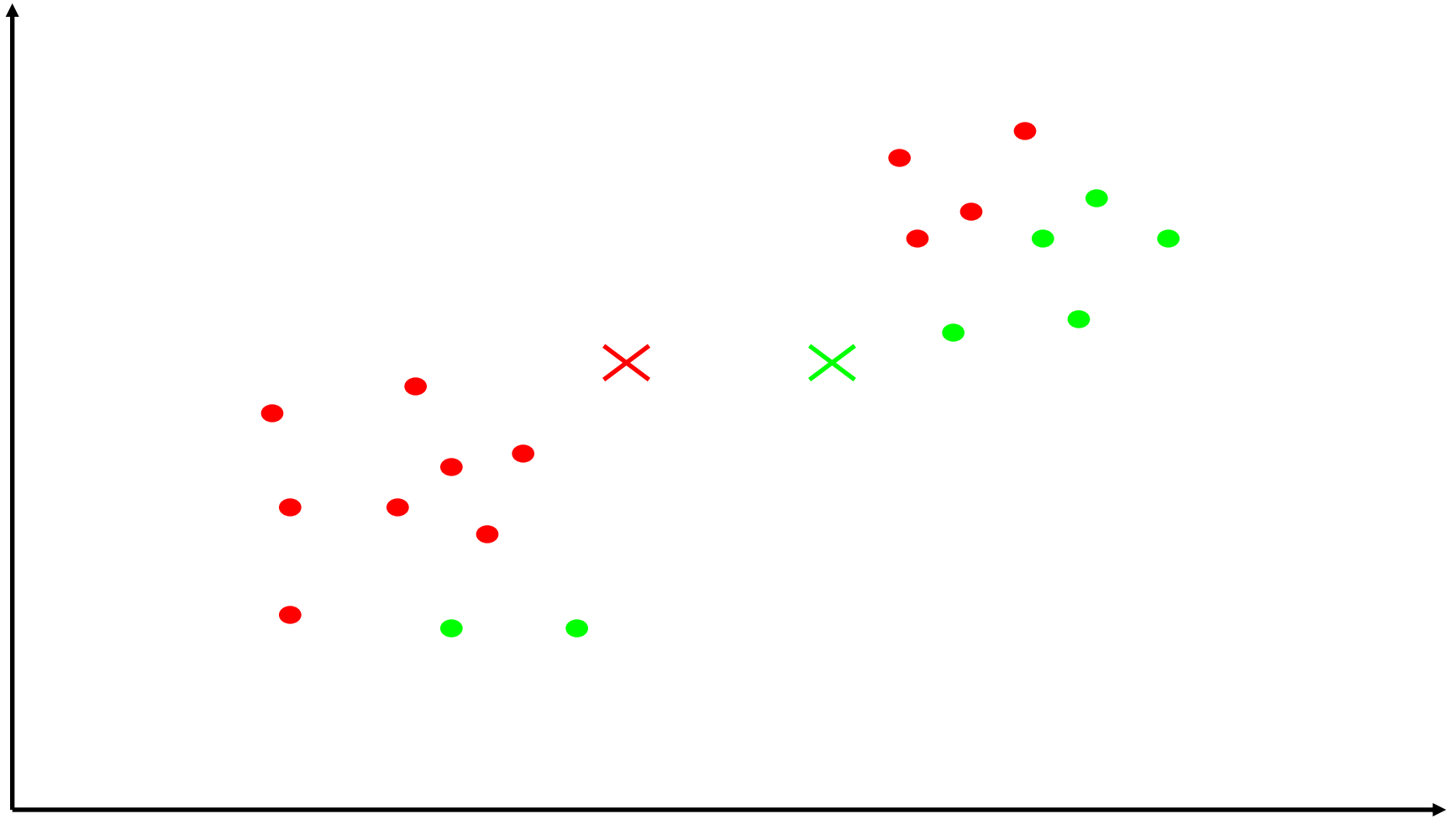
K-Means Clustering



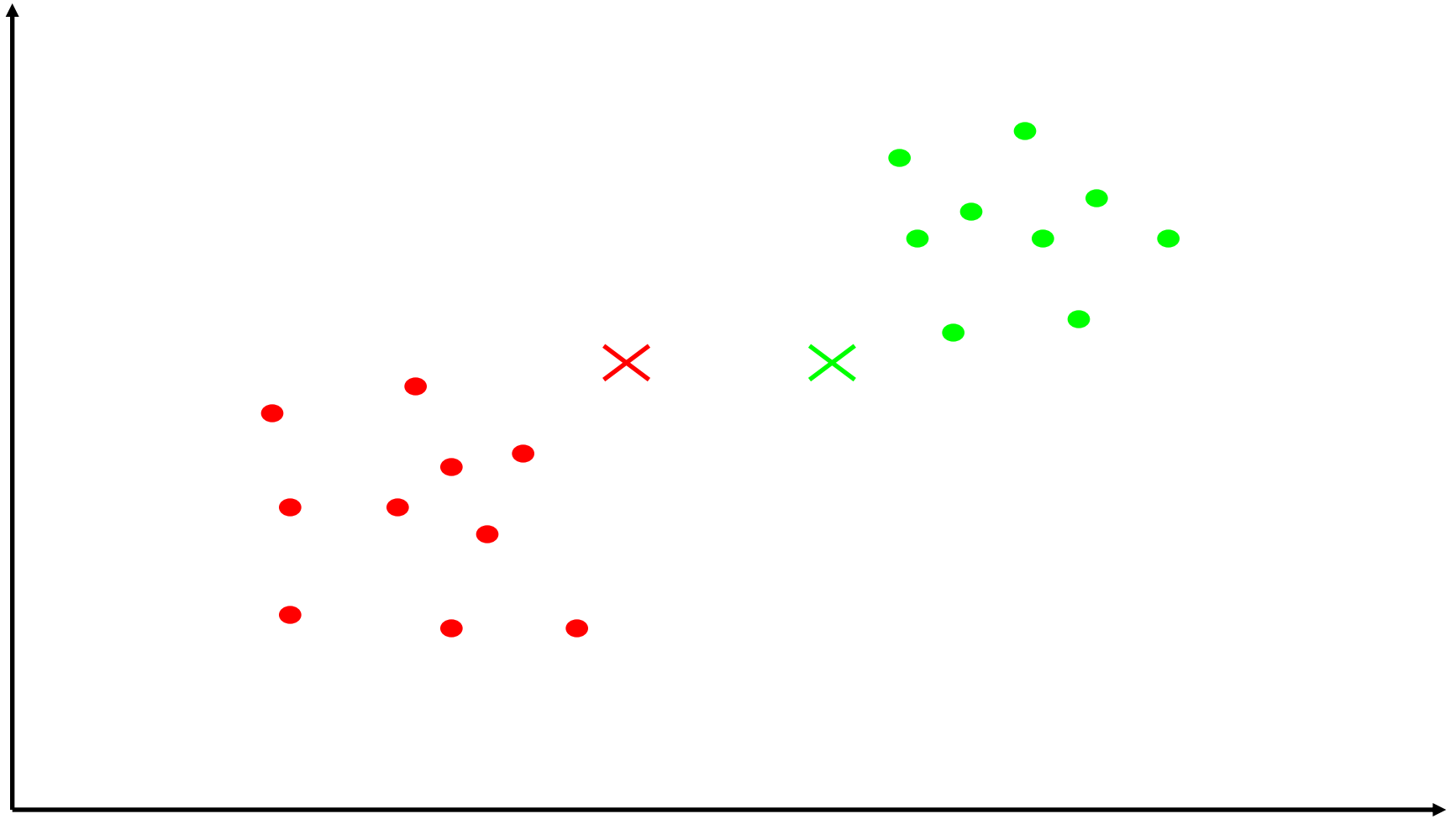
K-Means Clustering



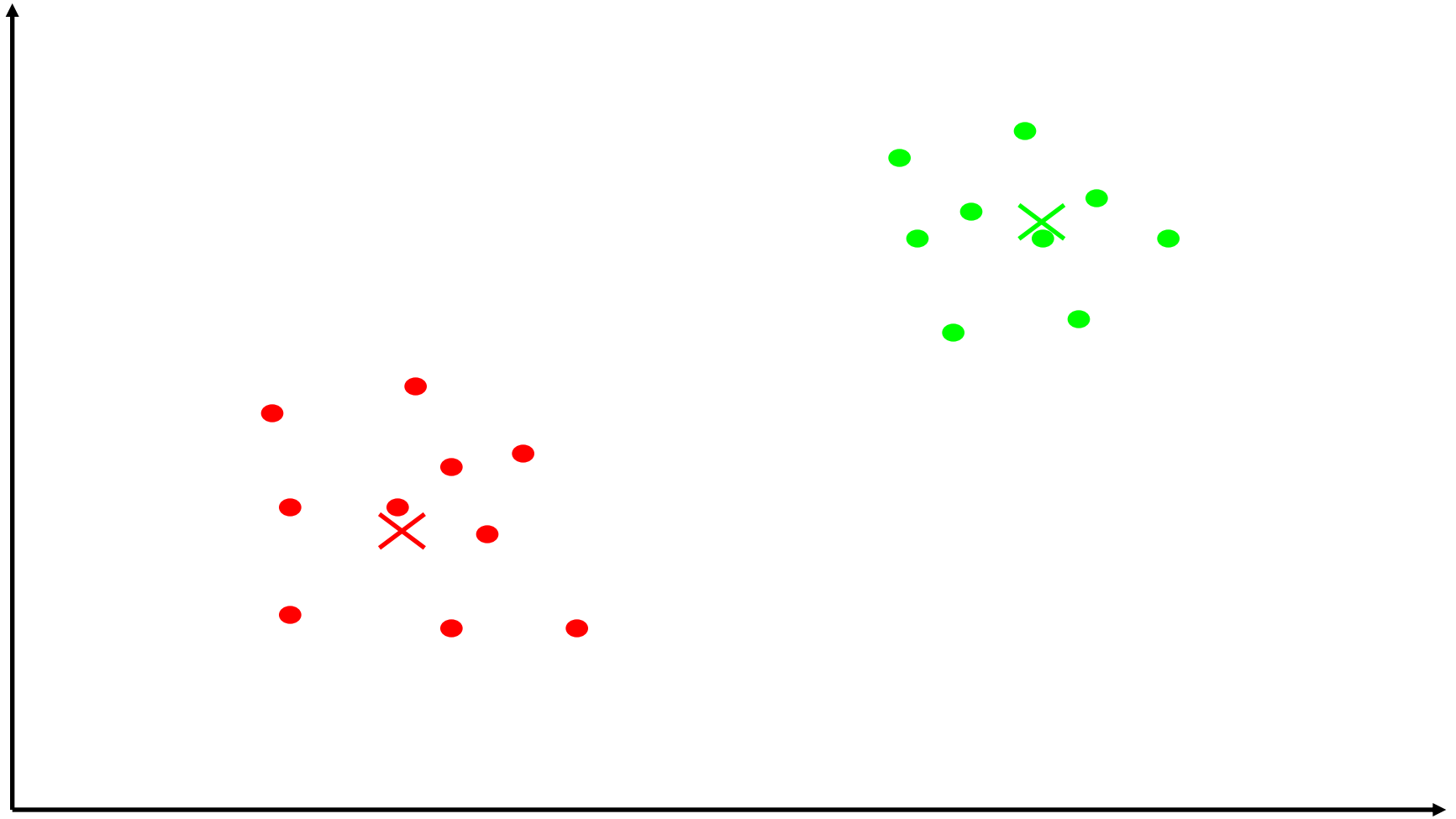
K-Means Clustering



K-Means Clustering



K-Means Clustering



k-means Clustering Algorithm

Initialization: Initial cluster center **w**(i); $1 \le i \le c$, D(–1)= 0, **I**(x_k,i) = 0, $1 \le i \le c$, $1 \le k \le K$;

Repeat

(A) Assign cluster membership (Expectation step)

Evaluate d(x_k, **w**(i)); $1 \le i \le c$, $1 \le k \le K$

I(x_k,i) = 1 if d(x_k, **w**(i)) < d(x_k, **w**(j)), $j \ne i$;

= 0; otherwise. $1 \le k \le K$

(B) Evaluate distortion D:

$$D(iter) = \sum_{k=1}^{N} I(x_k, i) d(x_k, w(i)) \quad 1 \le k \le K$$

(C) Update code words according to new assignment (Maximization)

$$W(i) = \sum_{k=1}^{N} I(x_k, i) x_k, \qquad N_i = \sum_{k=1}^{N} I(x_k, i), \qquad 1 \le i \le c$$

(D) Check for convergence

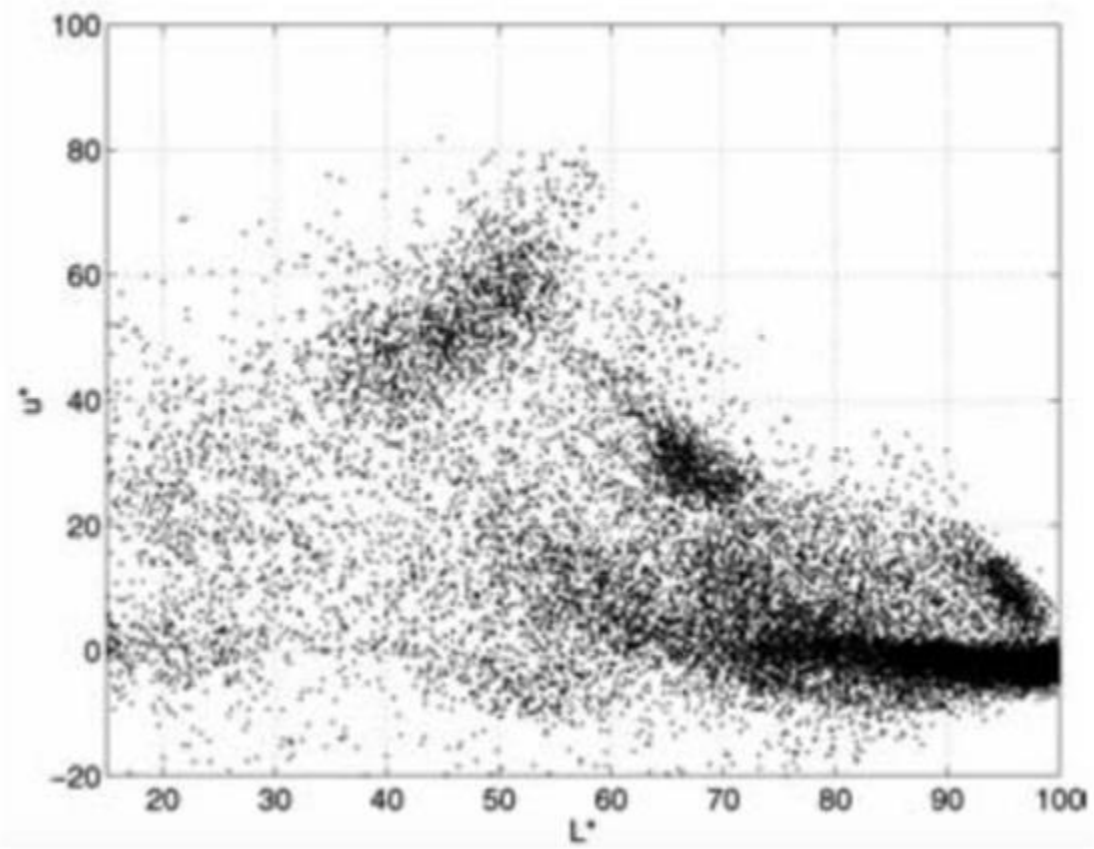
if 1–D(Iter–1)/D(Iter) < e , then convergent = TRUE,

Mean shift (non-parametric) segmentation

- Segmentation by clustering of the pixels in the image (colour and position)
- Non-parametric method (Parzen window technique) to find modes (i.e. peaks) in the density function
- All pixels climbing to the same peak are assigned to the same region.

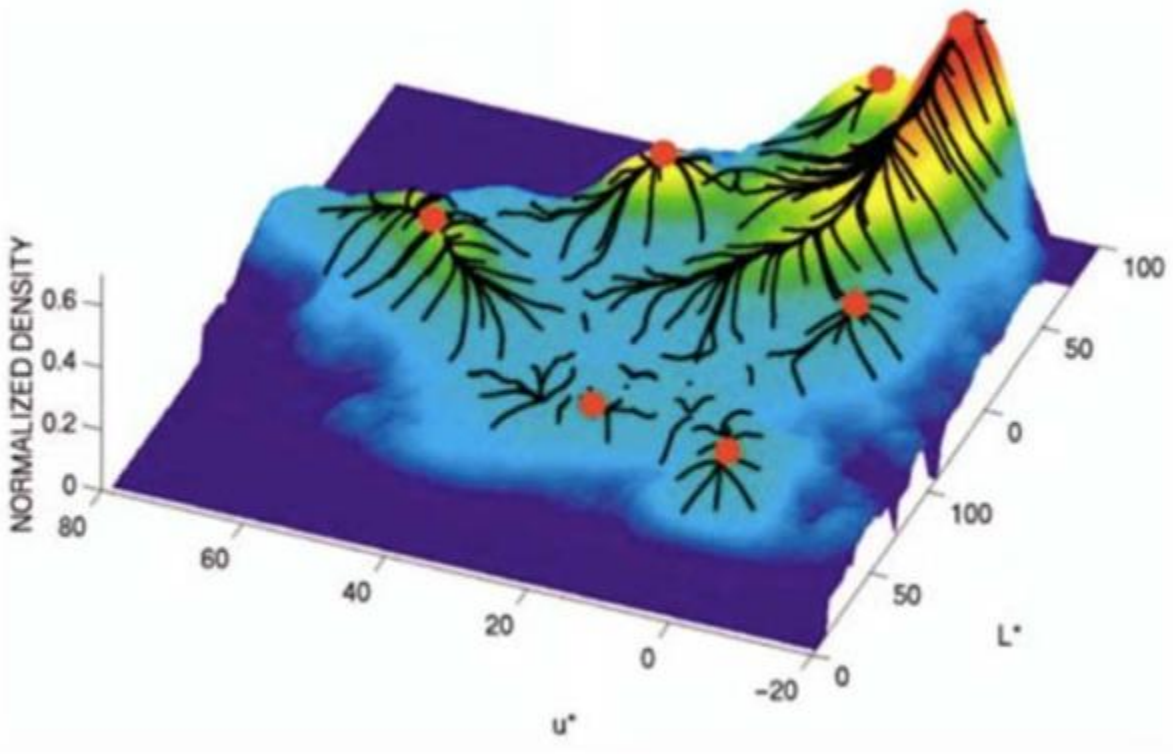
(Szeliski: Computer Vision - Algorithms and Applications)

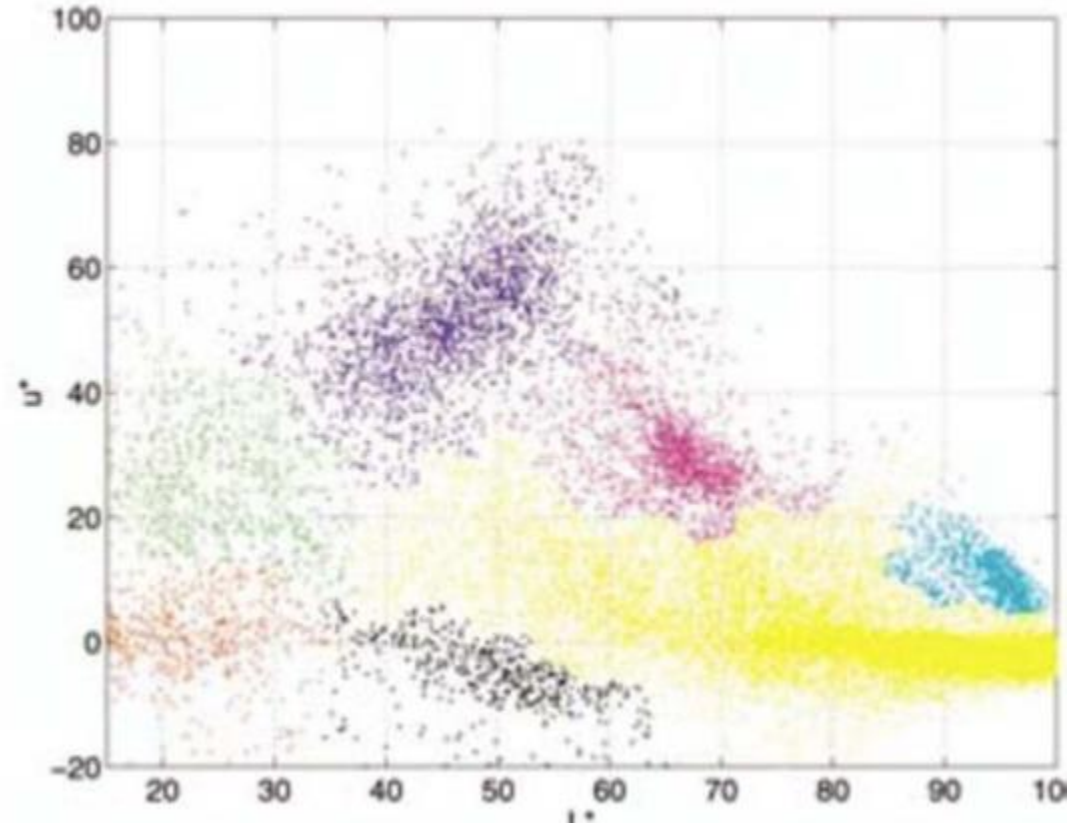
Mean shift segmentation



(Szeliski: Computer Vision - Algorithms and Applications)

Mean shift segmentation





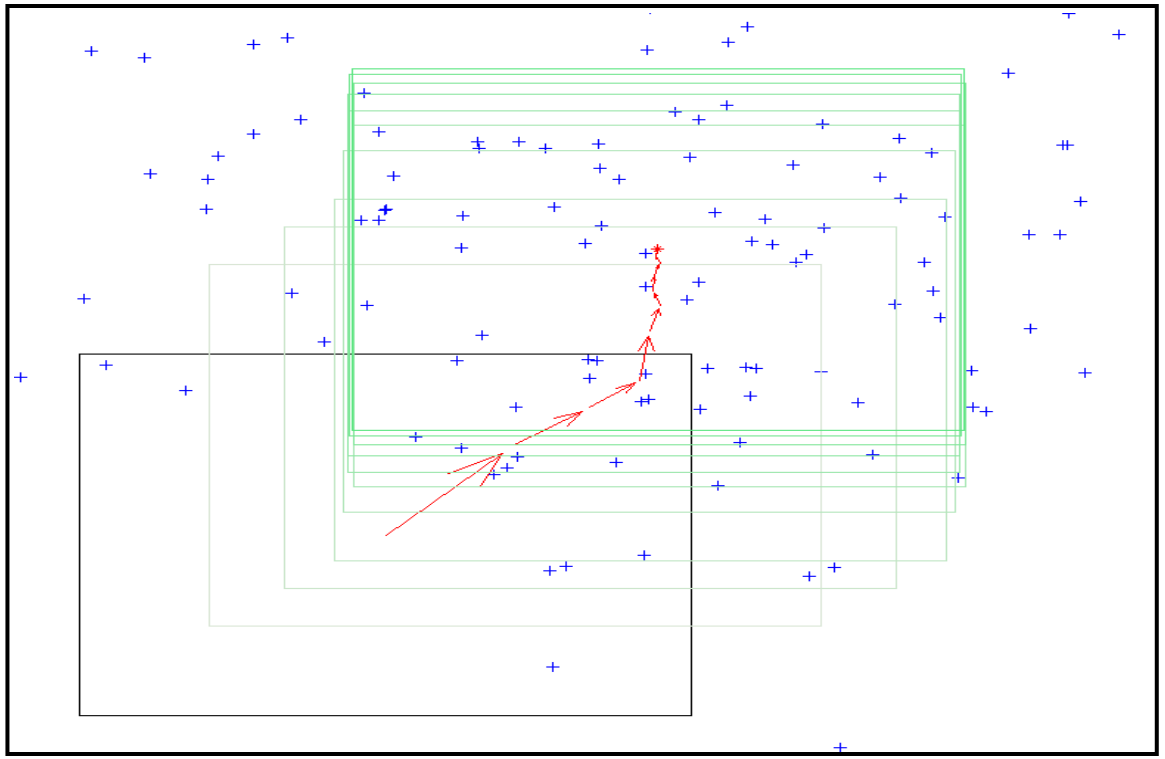
(Szeliski: Computer Vision - Algorithms and Applications)

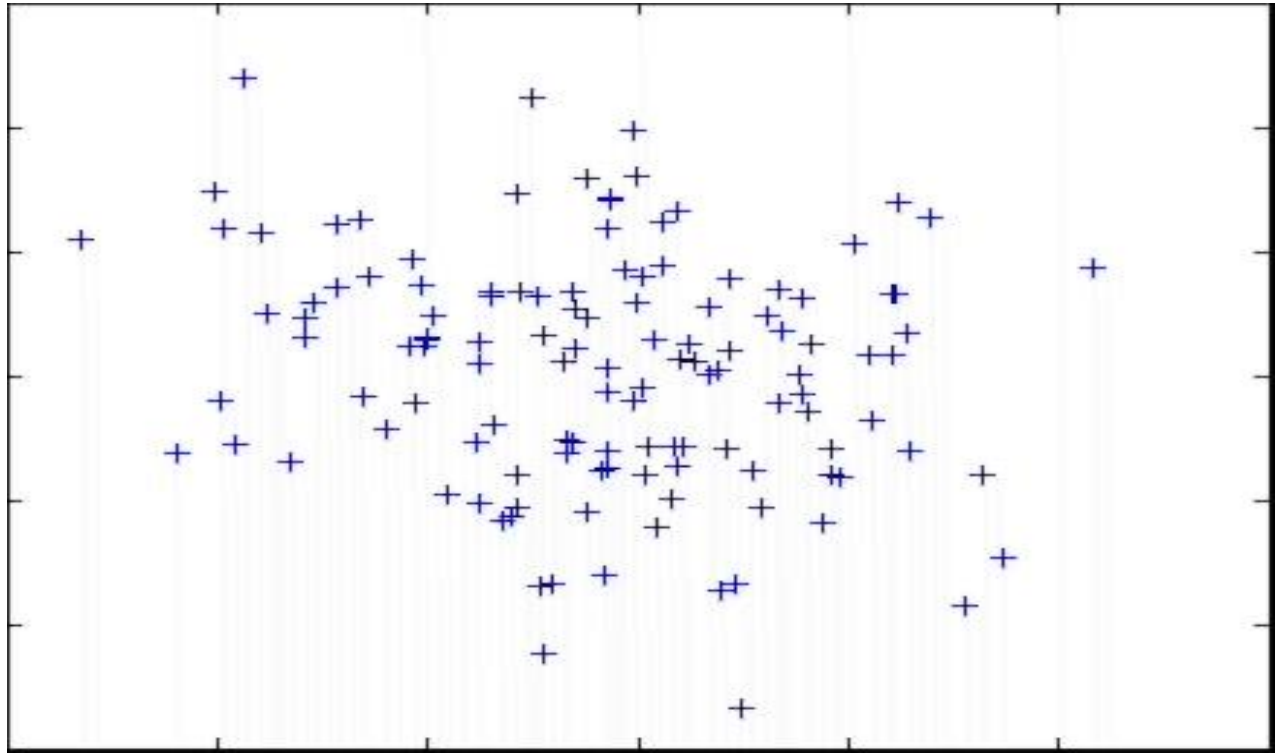
Mean Shift Algorithm

Mean Shift Algorithm

1. Choose a search window size.
2. Choose the initial location of the search window.
3. Compute the mean location (centroid of the data) in the search window.
4. Center the search window at the mean location computed in Step 3.
5. Repeat Steps 3 and 4 until convergence.

The mean shift algorithm seeks the “mode” or point of highest density of a data distribution:





Mean Shift Segmentation

Mean Shift Setmentation Algorithm

1. Convert the image into tokens (via color, gradients, texture measures etc).
2. Choose initial search window locations uniformly in the data.
3. Compute the mean shift window location for each initial position.
4. Merge windows that end up on the same “peak” or mode.
5. The data these merged windows traversed are clustered together.

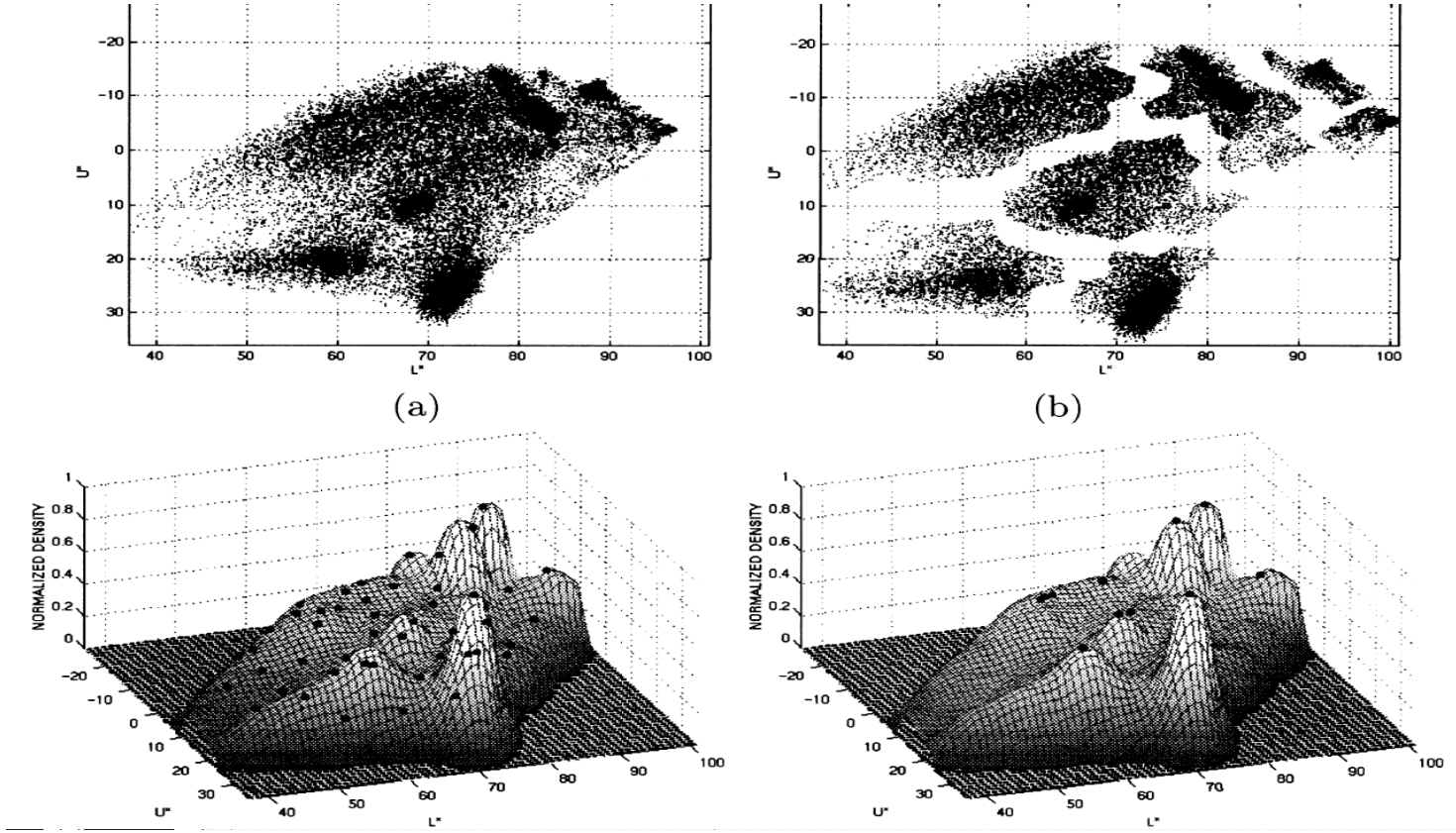


*Image From: Dorin Comaniciu and Peter Meer, Distribution Free Decomposition of Multivariate Data, Pattern Analysis & Applications (1999)2:22–30

Mean Shift Segmentation Extension

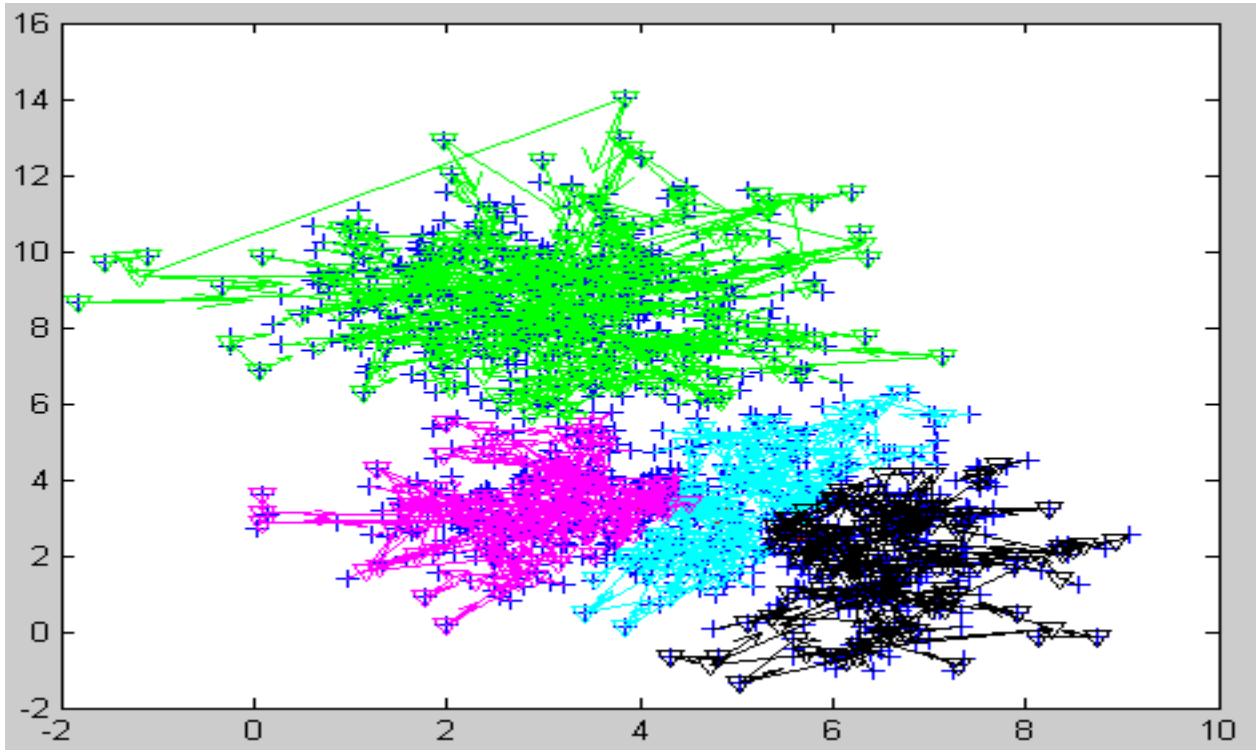
Is scale (search window size) sensitive. Solution, use all scales:

Gary Bradski's internally published agglomerative clustering extension:

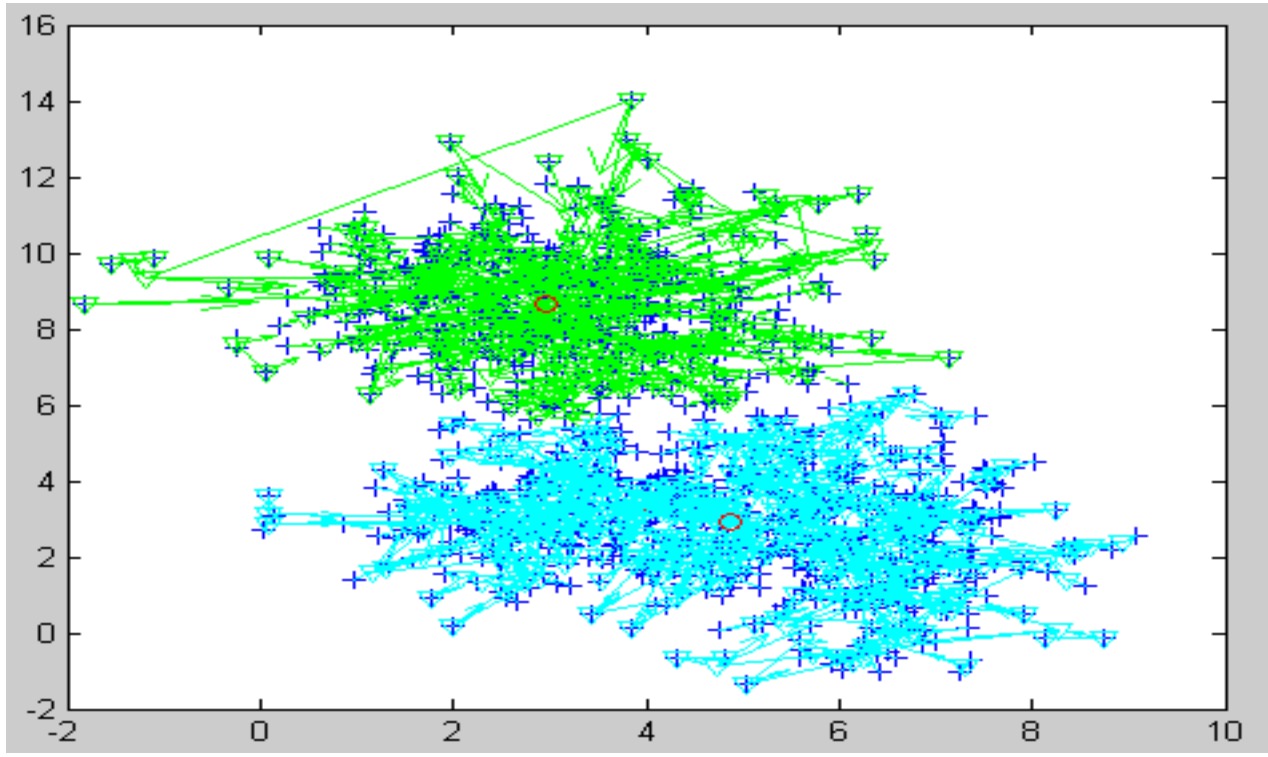
Mean shift dendrograms

1. Place a tiny mean shift window over each data point
2. Grow the window and mean shift it
3. Track windows that merge along with the data they transversed
4. Until everything is merged into one cluster

Best 4 clusters:

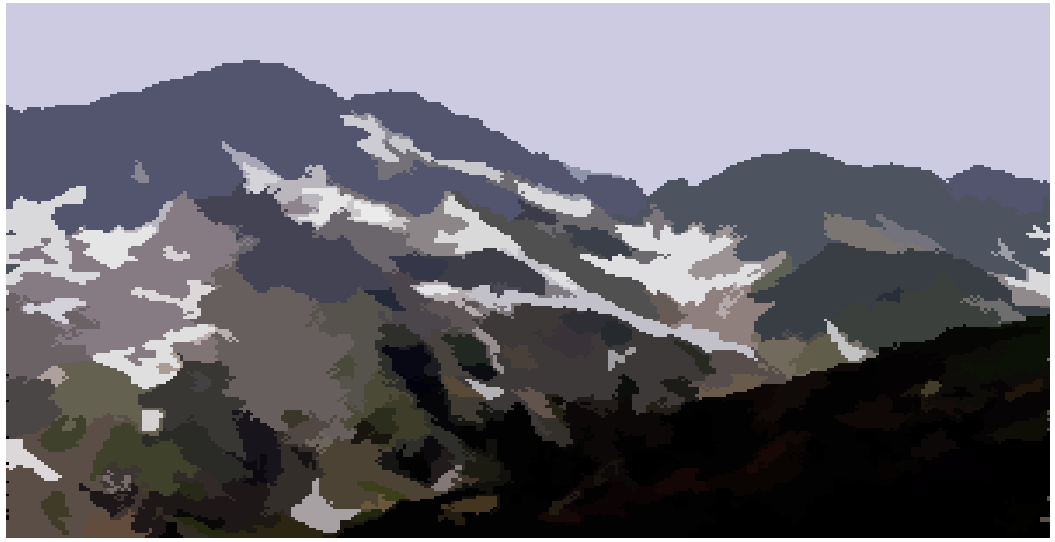


Best 2 clusters:



Advantage over agglomerative clustering: Highly parallelizable

Mean Shift Segmentation



http://www.caip.rutgers.edu/~comanici/MSPAMI/msPamiResults.html

Road segmentation for autonomous vehicles

Image data (graylevel, colour, local texture) from trapezoidal region is used to build a Gaussian model of the road surface.

Pixels with sufficiently high probability density with respect to the model are assigned to the road class (marked in green).

Road segmentation - alternative approach

Original RGB image converted to an illumination invariant colour space (reduced variation due to sunlight and shadows). From this image a local entropy image is derived (Matlab: entropyfilt).

Segmentation by region growing of the local entropy image (Matlab: grayconnected) using the green dots (left image) as seed pixels.

Morphological operations

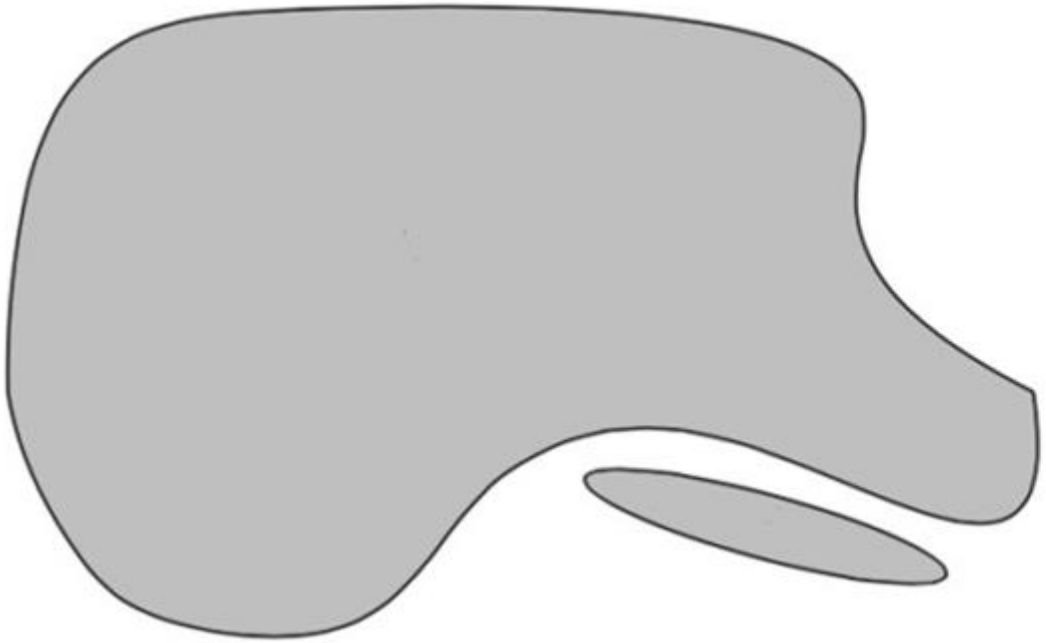
- Non-linear filtering
- Typically used to clean up binary images
- Erosion: replace pixel value with minimum in local neighborhood
- Dilation: replace pixel value with maximum in local neighborhood
- Structuring element used to define the local neighborhood:

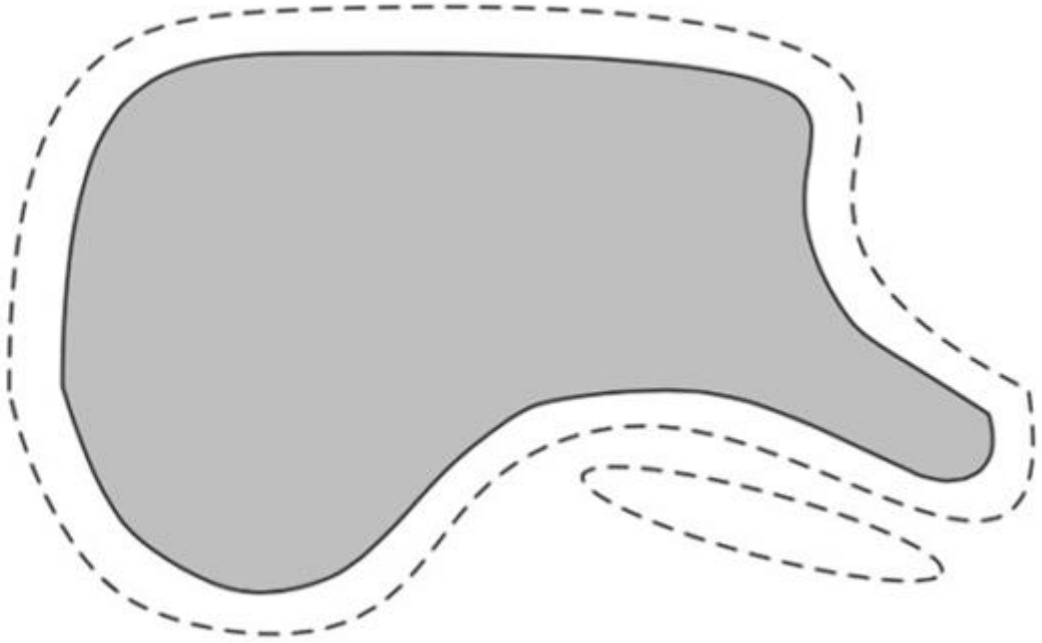
0	1	0
1	1	1
0	1	0

A shape (in blue) and its morphological dilation (in green) and erosion (in yellow) by a diamond-shape structuring element.

Morphological operations - Erosion

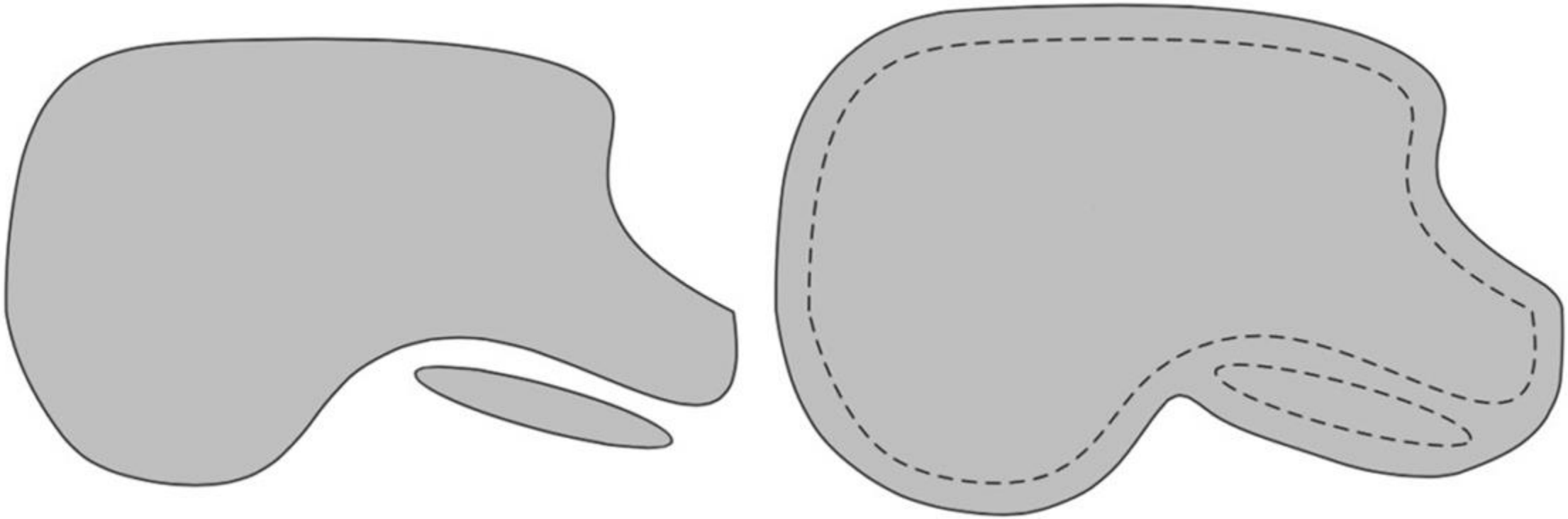
Structuring element (disk shaped)



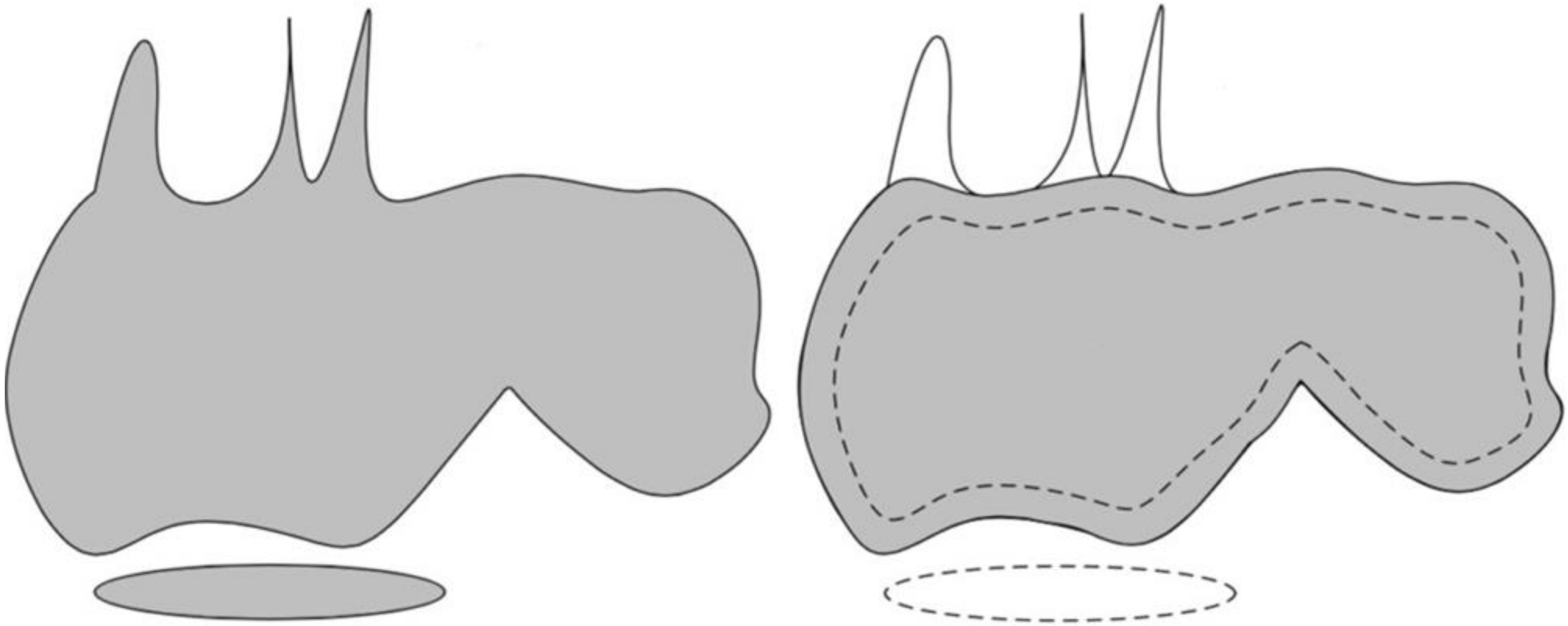


Morphological operations - Dilation

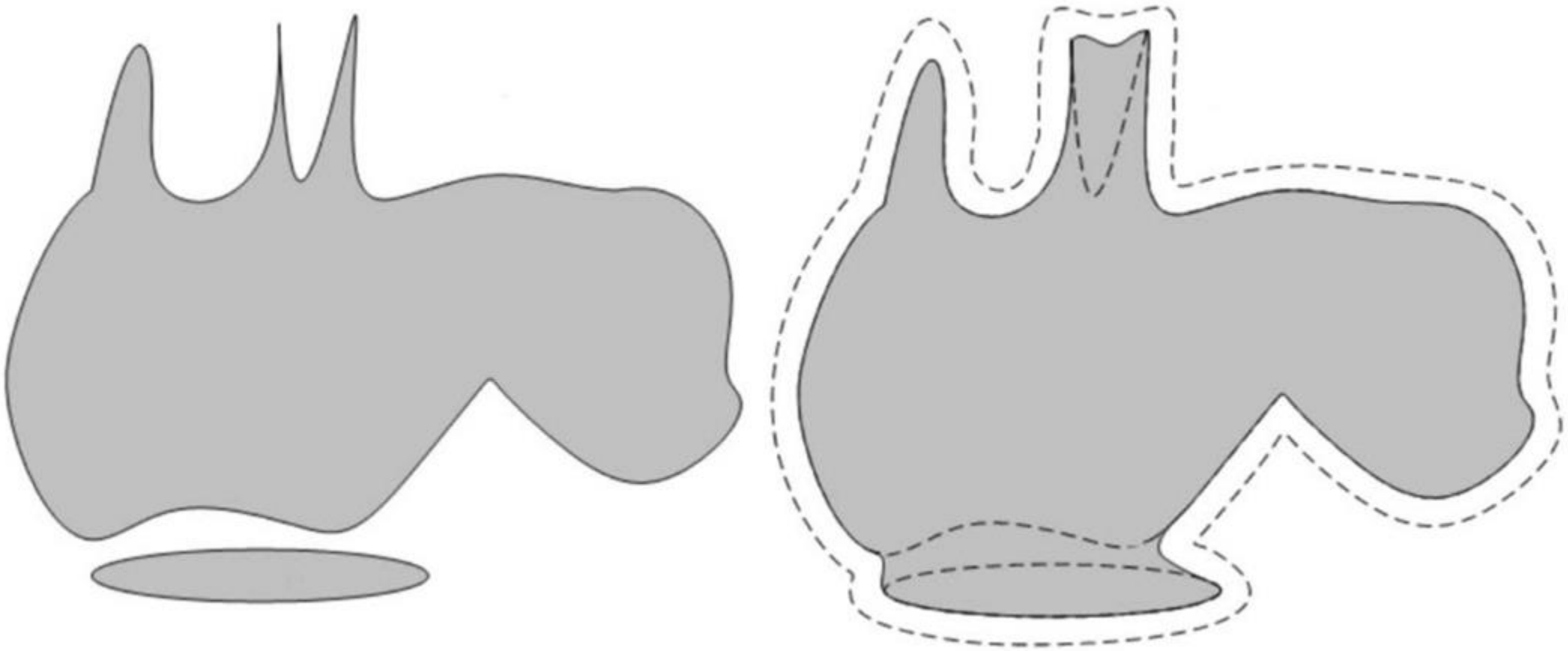
Structuring element (disk shaped)



Opening = Erosion + Dilation



Closing = Dilation + Erosion



Opening - example

Thresholded image

Result of opening

Disk shaped structuring element with radius = 2 pixels (5 x 5 filter mask)

Closing - example

Thresholded image

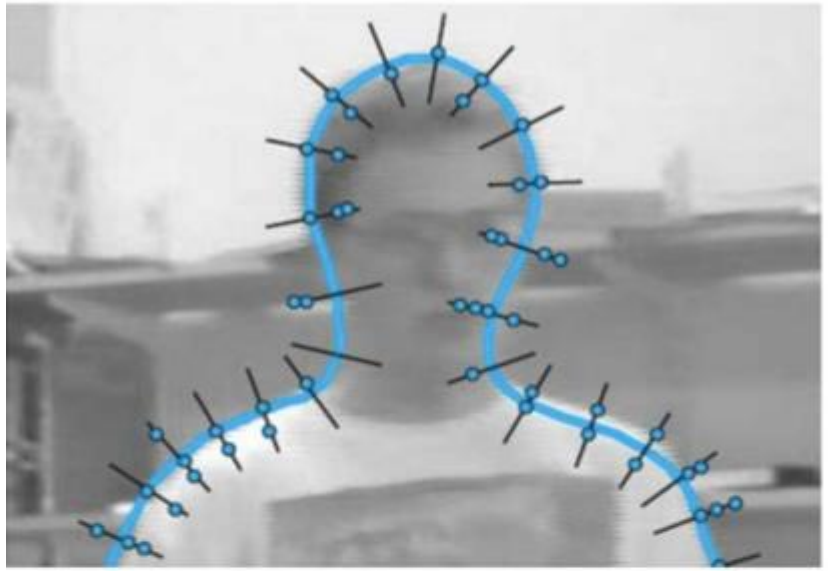
Result of opening

Disk shaped structuring element with radius = 2 pixels (5 x 5 filter mask)

Active contours

Fitting of curves to object boundaries:

- Snakes (fitting of spline curves to strong edges)
- Intelligent scissors (interactive specification of curves clinging to object boundaries)
- Level set techniques (evolving boundaries as the zero set of a characteristic function).



(Szeliski: Computer Vision - Algorithms and Applications)

Split and merge methods

Principles:

- Recursive splitting of the image based on region statistics
- Hierarchical merging of pixels and regions
- Combined splitting and merging

Methods:

- Watershed segmentation
- Region splitting (divisive clustering)
- Region merging (agglomerative clustering)
- Graph-based segmentation

(Szeliski: Computer Vision - Algorithms and Applications)

Split

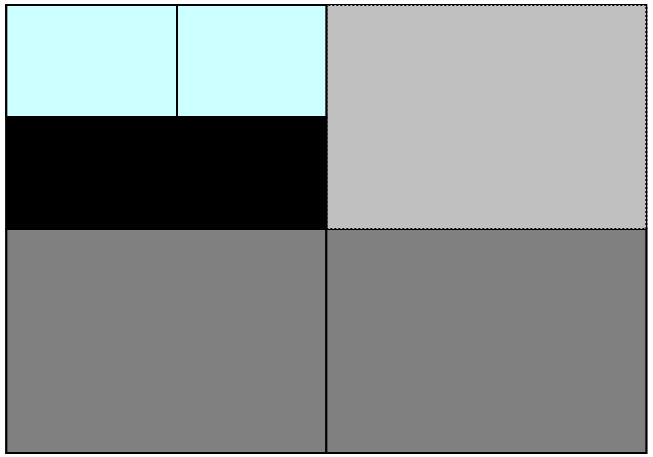
- The opposite approach to region growing is region splitting.
- It is a top-down approach and it starts with the assumption that the entire image is homogeneous
- If this is not true, the image is split into four sub images
- This splitting procedure is repeated recursively until we split the image into homogeneous regions

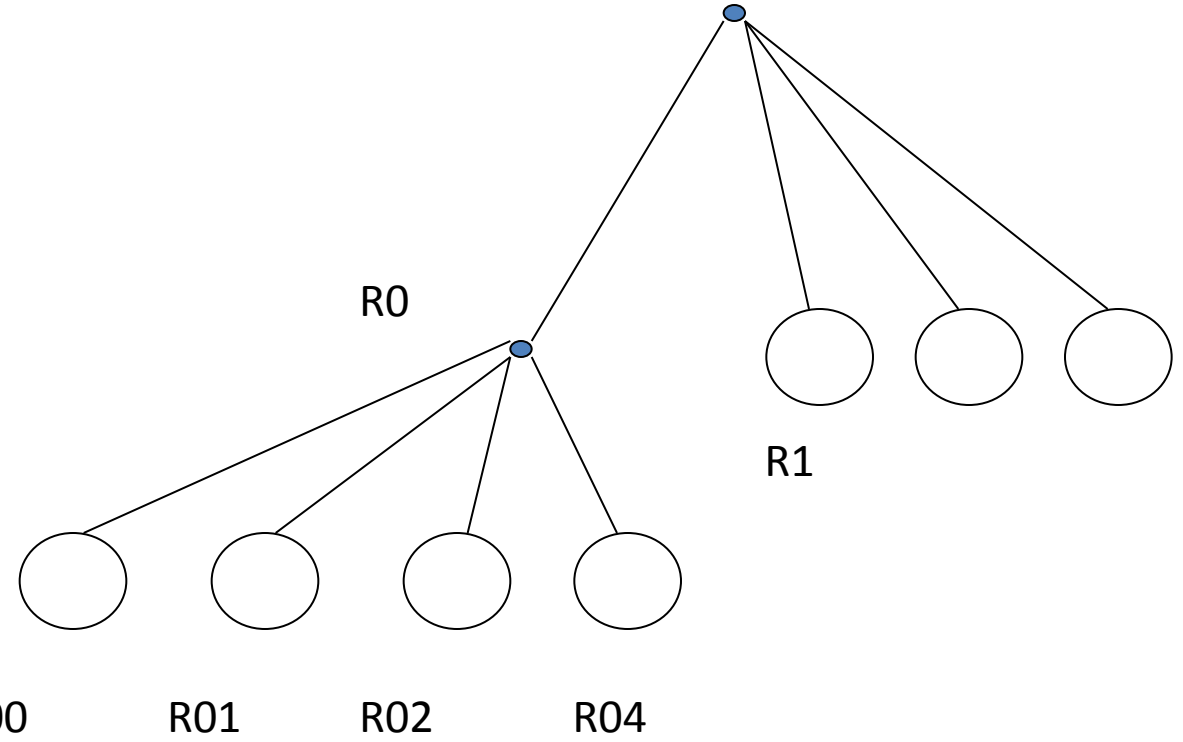
Split

- If the original image is square N x N, having dimensions that are powers of 2(N = 2n)
- All regions produced but the splitting algorithm are squares having dimensions M x M , where M is a power of 2 as well.
- Since the procedure is recursive, it produces an image representation that can be described by a tree whose nodes have four sons each
- Such a tree is called a Quadtree.

Split

Quadtree





Split

- Splitting techniques disadvantage, they create regions that may be adjacent and homogeneous, but not merged.

- Split and Merge method is an iterative algorithm that includes both splitting and merging at each iteration:

Split / Merge

- If a region R is inhomogeneous (P(R)= False) then is split into four sub regions

- If two adjacent regions Ri,Rj are homogeneous (P(Ri U Rj) = TRUE), they are merged

- The algorithm stops when no further splitting or merging is possible

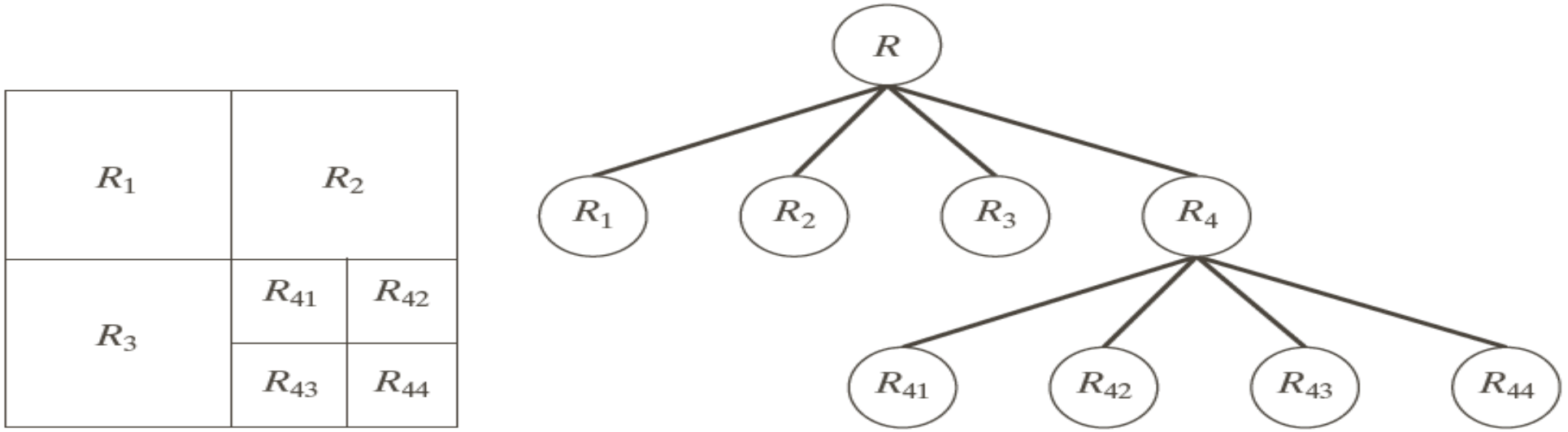
The split and merge algorithm produces more compact regions than the pure splitting algorithm

Region Splitting and Merging

R: entire image R_i: entire image Q: predicate

1. For any region R_i, If $Q(R_i) = \text{FALSE}$,
   we divide the image R_i into quadrants.
2. When no further splitting is possible,
   merge any adjacent regions R_j and R_k
   for which $Q(R_j \cup R_k) = \text{TRUE}$.
3. Stop when no further merging is possible.

Region Splitting and Merging

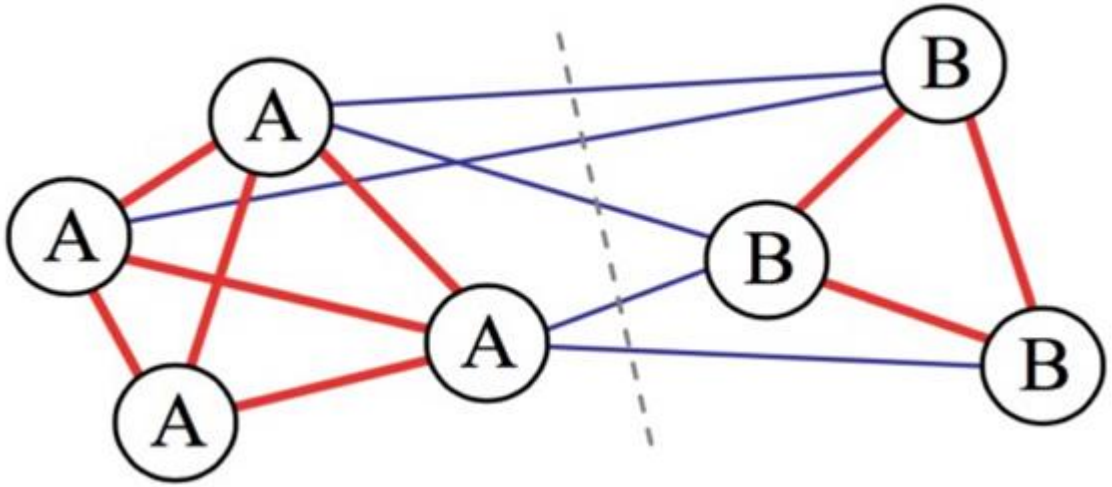


a b

FIGURE 10.52 (a) Partitioned image. (b) Corresponding quadtree. R represents the entire image region.

Normalized cuts

Separation of groups with weak affinities (similarities) between nearby pixels



(Szeliski: Computer Vision - Algorithms and Applications)

Graph cuts

(Szeliski: Computer Vision - Algorithms and Applications)

Energy-based methods for binary segmentation:

- Grouping of pixels with similar statistics
- Minimization of pixel-based energy function
- Region-based and boundary-based energy terms
- Image represented as a graph
- Cutting of weak edges, i.e. low similarity between corresponding pixels.